МИНИСТЕРСТВО ОБРАЗОВАНИЯ РОСТОВСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССТОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ РОСТОВСКОЙ ОБЛАСТИ
«РОСТОВСКИЙ-НА-ДОНУ КОЛЛЕДЖ СВЯЗИ И ИНФОРМАТИКИ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по профессиональному модулю

ПМ.03 Обучение готовых моделей искусственного интеллекта
по специальности
09.02.13 Интеграция решений с применением технологий искусственного интеллекта
(базовой подготовки)

г. Ростов-на-Дону, 2026 г.

РАССМОТРЕНО
на заседании цикловой комиссии
Программирование
Протокол № 1 от 31.08.2026 года
Председатель ЦК Сулавко С.Н.

УТВЕРЖДАЮ:
Зам. директора по УМР
И.В. Подцатова
«16» ноября 2026 г.

Комплект контрольно-оценочных средств по ПМ.03 «Обучение готовых моделей искусственного интеллекта» разработан в соответствии с рабочей программой по ПМ, разработанной по специальности 09.02.13 Интеграция решений с применением технологий искусственного интеллекта.

Организация-разработчик: государственное бюджетное профессиональное образовательное учреждение Ростовской области «Ростовский-на-Дону колледж связи и информатики»

Разработчики:

Трифанкова А.А., Черевко А.Д., Кошкина А.А., Сулавко С.Н преподаватели государственного бюджетного профессионального образовательного учреждения Ростовской области «Ростовский-на-Дону колледж связи и информатики»

Рецензент:

СОДЕРЖАНИЕ

1. Формы промежуточной аттестации по профессиональному модулю.....4
2. Результаты освоения модуля, подлежащие проверке …….4
3. Оценка освоения теоретического курса профессионального модуля 13
4. Контроль приобретения практического опыта. Оценка по учебной и (или) производственной практике.....254
5. Контрольно-оценочные материалы для квалификационного экзамена ……260

Общие положения

Результатом освоения профессионального модуля является готовность обучающегося к выполнению вида профессиональной деятельности (в соответствии с рабочей программой ПМ) и сформированность профессиональных и общих компетенций.

Формой аттестации по профессиональному модулю является экзамен по модулю.

1. Формы промежуточной аттестации по профессиональному модулю

Таблица 1

Элементы модуля, профессиональный модуль	Формы промежуточной аттестации
1	**2**
МДК 03.01 Разработка сценариев обучения готовых моделей	Экзамен
МДК 03.02 Интеграция искусственного интеллекта в информационные системы	Дифференцированный зачет
МДК 03.03 Разработка промптов для искусственного интеллекта	Дифференцированный зачет
МДК 03.04 Управление жизненным циклом моделей ИИ	Дифференцированный зачет
УП.03	Зачет
ПП.03	Зачет
ПМ 03	**Экзамен по модулю**

2. Результаты освоения модуля, подлежащие проверке

2.1. Профессиональные и общие компетенции

В результате контроля и оценки по профессиональному модулю осуществляется комплексная проверка следующих профессиональных и общих компетенций:

Таблица 2

Профессиональные компетенции	Показатели оценки результата
ПК 3.1	- Подбор и настройка готовых моделей ИИ с учетом поставленных задач, анализ результатов их применения
ПК 3.2	- Создание сценариев обучения, подготовка данных для обучения, настройка гиперпараметров для достижения оптимального результата
ПК 3.3	- Обучение моделей на подготовленных данных, применение методов калибровки для улучшения точности моделей
ПК 3.4	- Оценка эффективности обученных моделей, корректировка обучения при необходимости, анализ ошибок и улучшение модели
ПК 3.5	- Создание отчетов по обучению моделей, использование инструментов для визуализации (Matplotlib,

	Seaborn) для наглядного представления данных
ПК 3.6	- Формирование запросов для получения и анализа данных, построение графиков и диаграмм для визуализации результатов работы ИИ
ПК 3.7	- Определение источников больших данных для анализа, идентификация внешних и внутренних источников данных для проведения аналитических работ - Получение и фильтрация больших объемов данных из гетерогенных источников - Извлечение, проверка и очистка больших объемов данных из гетерогенных источников - Агрегация и разработка представления больших объемов данных из гетерогенных источников - Оценка соответствия набора данных предметной области и задачам аналитических работ - Выбор методов и инструментальных средств анализа больших данных для проведения аналитических работ - Разработка, проверка и оценка используемых моделей больших данных - Адаптация и развертывание моделей больших данных в предметной среде - Выбор средств представления результатов аналитики больших данных - Подготовка отчета по результатам аналитических работ
ПК 3.8	- Разработка кода ИС и баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Верификация кода ИС и баз данных ИС относительно дизайна ИС и структуры баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Устранение обнаруженных несоответствий в коде ИС и базах данных ИС относительно дизайна ИС и структуры баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разработка интерфейсов обмена данными между ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разработка форматов обмена данными между ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разработка технологий обмена данными между ИС заказчика в соответствии с трудовым заданием в рамках выполнения работ по созданию (модификации) и сопровождению ИС
ПК 3.9	- Переработка текстов различной тематики (рерайт) - Поисковая оптимизация и адаптация текстовых материалов
ПК 3.10	- Создание программного кода в соответствии с техническим заданием (готовыми спецификациями) - Оптимизация программного кода с использованием специализированных программных средств

ПК 3.11	- Сбор идей о требованиях к ИТ-продукту из различных источников - Разработка функциональных требований к ИТ-продукту - Создание макетов интерфейса ИТ-продукта - Разработка нефункциональных требований к ИТ-продукту
ПК 3.12	- Организация разработки и разработка плана управления требованиями заказчика в проектах в области ИТ любого уровня сложности - Согласование плана управления требованиями в проектах в области ИТ любого уровня сложности с заинтересованными лицами Утверждение плана управления требованиями заказчика в проектах в области ИТ любого уровня сложности

Таблица 3

Общие компетенции	Показатели оценки результата
ОК 01	- Выбор способов решения задач профессиональной деятельности применительно к различным контекстам
ОК 02	- Использование современных средств поиска, анализа и интерпретации информации, и информационные технологии
ОК 05	- Осуществление устной и письменной коммуникации на государственном языке Российской Федерации
ОК 09	- Использование профессиональной документацией на государственном и иностранном языках

2.2. Общие и (или) профессиональные компетенции, проверяемые дополнительно: _____

2.3. Требования к портфолио

Тип портфолио: портфолио работ

Выберите из предложенных вариантов: портфолио документов, портфолио работ, портфолио смешанного типа.

Проверяемые результаты обучения:

Основные требования

Требования к структуре и оформлению портфолио:

1. В портфолио входят практические работы, входящие в МДК.03.01, МДК 03.02, МДК 03.03 и МДК 03.04 отзывы руководителей учебной и производственной практик
2. Практические работы оформлены в виде отчета.
3. Оформление отчета соответствует требованиям документа «Правила оформления текстовых документов в учебном процессе»

Показатели оценки портфолио

Таблица 4

Коды и наименования проверяемых компетенций или их сочетаний	Показатели оценки результата	Оценка (да / нет)
ОК 01	- Выбор способов решения задач профессиональной деятельности применительно к различным контекстам	
ОК 02	- Использование современных средств поиска, анализа и интерпретации информации, и информационные технологии	
ОК 05	- Осуществление устной и письменной коммуникации на государственном языке Российской Федерации	
ОК 09	- Использование профессиональной документацией на государственном и иностранном языках	
ПК 3.1	- Подбор и настройка готовых моделей ИИ с учетом поставленных задач, анализ результатов их применения	
ПК 3.2	- Создание сценариев обучения, подготовка данных для обучения, настройка гиперпараметров для достижения оптимального результата	
ПК 3.3	- Обучение моделей на подготовленных данных, применение методов калибровки для улучшения точности моделей	
ПК 3.4	- Оценка эффективности обученных моделей, корректировка обучения при необходимости, анализ ошибок и улучшение модели	
ПК 3.5	- Создание отчетов по обучению моделей, использование инструментов для визуализации (Matplotlib, Seaborn) для наглядного представления данных	
ПК 3.6	- Формирование запросов для получения и анализа данных, построение графиков и диаграмм для визуализации результатов работы ИИ	
ПК 3.7	- Определение источников больших данных для анализа, идентификация внешних и внутренних источников данных для проведения аналитических работ - Получение и фильтрация больших объемов данных из гетерогенных источников - Извлечение, проверка и очистка больших объемов данных из гетерогенных источников - Агрегация и разработка представления больших объемов данных из гетерогенных источников - Оценка соответствия набора данных предметной области и задачам аналитических работ	

	- Выбор методов и инструментальных средств анализа больших данных для проведения аналитических работ - Разработка, проверка и оценка используемых моделей больших данных - Адаптация и развертывание моделей больших данных в предметной среде - Выбор средств представления результатов аналитики больших данных - Подготовка отчета по результатам аналитических работ	
ПК 3.8	- Разработка кода ИС и баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Верификация кода ИС и баз данных ИС относительно дизайна ИС и структуры баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Устранение обнаруженных несоответствий в коде ИС и базах данных ИС относительно дизайна ИС и структуры баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разработка интерфейсов обмена данными между ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разработка форматов обмена данными между ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разработка технологий обмена данными между ИС заказчика в соответствии с трудовым заданием в рамках выполнения работ по созданию (модификации) и сопровождению ИС	
ПК 3.9	- Переработка текстов различной тематики (рерайт) - Поисковая оптимизация и адаптация текстовых материалов	
ПК 3.10	- Создание программного кода в соответствии с техническим заданием (готовыми спецификациями) - Оптимизация программного кода с использованием специализированных программных средств	
ПК 3.11	- Сбор идей о требованиях к ИТ-продукту из различных источников - Разработка функциональных требований к ИТ-продукту - Создание макетов интерфейса ИТ-продукта - Разработка нефункциональных требований к ИТ-продукту	
ПК 3.12	- Организация разработки и разработка плана управления требованиями заказчика в проектах в области ИТ любого уровня сложности	

	- Согласование плана управления требованиями в проектах в области ИТ любого уровня сложности с заинтересованными лицами - Утверждение плана управления требованиями заказчика в проектах в области ИТ любого уровня сложности	

2.4 Требования к курсовой работе (проекту) как части экзамена по модулю

Проверяемые результаты обучения:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста.

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта.

ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта.

ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта.

ПК 3.5. Оформлять результат проведения процедуры обучения.

ПК 3.6. Формировать запросы для работы с искусственным интеллектом с целью визуализации данных.

ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта.

Задание на курсовой проект

- Сформулировать проблему, цель, критерий успеха и ожидаемый результат; обосновать выбор типа задачи машинного обучения.
- Описать источник, состав, объём и качество данных; провести первичный анализ, очистку, обработку пропусков и преобразование признаков.
- Организовать корректное разделение на обучающую, валидационную и тестовую выборки (при необходимости применить кросс-валидацию), исключив утечку данных.
- Выбрать не менее одной готовой модели и базовый вариант для сравнения; построить сценарий обучения, дообучения или настройки гиперпараметров.
- Проверить качество модели подходящими метриками, визуализировать результаты и проанализировать ошибки и ограничения.

- Создать демонстрационный программный сценарий применения модели: интерфейс, API, пакетный обработчик либо иной воспроизводимый прототип.
- Провести функциональное тестирование; рассмотреть вопросы этики, безопасности, правомерности использования данных и модели.
- Подготовить пояснительную записку, исходный код, инструкцию запуска, демонстрацию и материалы защиты.

Ожидаемый практический результат: воспроизводимый проект с подготовленными данными или ссылкой на разрешённый источник, кодом сценария, сохранённой моделью или описанием её получения, отчётом по метрикам и работоспособной демонстрацией. Использование готовой модели обязательно; разработка нового алгоритма «с нуля» не требуется.

Рекомендуемая структура курсового проекта

- Титульный лист (по форме образовательной организации).
- Индивидуальное задание на курсовой проект.
- Содержание с указанием страниц.
- Введение: актуальность, проблема, цель, задачи, объект и предмет проектирования, ожидаемый результат.
- Глава 1. Аналитическая часть: предметная область, постановка задачи, обзор готовых моделей и инструментов, характеристика данных, обоснование выбора.
- Глава 2. Проектно-практическая часть: архитектура решения, подготовка данных, сценарий обучения и настройки, реализация прототипа и интеграции.
- Глава 3. Тестирование и оценка: методика эксперимента, метрики, сравнительные результаты, визуализации, анализ ошибок, ограничения и вопросы безопасного применения.
- Заключение: достигнутые результаты, соответствие цели и задачам, перспективы доработки.
- Список использованных источников с описанием наборов данных, моделей и библиотек.
- Приложения: существенные фрагменты кода, инструкции запуска, схемы, таблицы, дополнительные графики, скриншоты и протоколы испытаний.

Пояснительная записка должна содержать ссылки на рисунки, таблицы, используемые источники и приложения. Код целесообразно передавать отдельным архивом или ссылкой на доступный репозиторий; большие листинги не следует помещать в основную часть. Объём записки и параметры оформления устанавливаются локальными методическими указаниями.

Показатели оценки работы (проекта)

Таблица 5

Коды и наименования проверяемых компетенций или их сочетаний	Показатели оценки результата	Оценка (да / нет)
ОК 01	- Выбор способов решения задач профессиональной деятельности применительно к различным контекстам	
ОК 02	- Использование современных средств поиска, анализа и интерпретации информации, и информационные технологии	
ОК 05	- Осуществление устной и письменной коммуникации на государственном языке Российской Федерации	
ОК 09	- Использование профессиональной документацией на государственном и иностранном языках	
ПК 3.1	- Подбор и настройка готовых моделей ИИ с учетом поставленных задач, анализ результатов их применения	
ПК 3.2	- Создание сценариев обучения, подготовка данных для обучения, настройка гиперпараметров для достижения оптимального результата	
ПК 3.3	- Обучение моделей на подготовленных данных, применение методов калибровки для улучшения точности моделей	
ПК 3.4	- Оценка эффективности обученных моделей, корректировка обучения при необходимости, анализ ошибок и улучшение модели	
ПК 3.5	- Создание отчетов по обучению моделей, использование инструментов для визуализации (Matplotlib, Seaborn) для наглядного представления данных	
ПК 3.6	- Формирование запросов для получения и анализа данных, построение графиков и диаграмм для визуализации результатов работы ИИ	
ПК 3.7	- Определение источников больших данных для анализа, идентификация внешних и внутренних источников данных для проведения аналитических работ - Получение и фильтрация больших объемов данных из гетерогенных источников - Извлечение, проверка и очистка больших объемов данных из гетерогенных источников - Агрегация и разработка представления больших объемов данных из гетерогенных источников - Оценка соответствия набора данных предметной области и задачам аналитических работ	

	- Выбор методов и инструментальных средств анализа больших данных для проведения аналитических работ - Разработка, проверка и оценка используемых моделей больших данных - Адаптация и развертывание моделей больших данных в предметной среде - Выбор средств представления результатов аналитики больших данных - Подготовка отчета по результатам аналитических работ	

Показатели оценки защиты работы (проекта)

Таблица 6

Коды и наименования проверяемых компетенций или их сочетаний	**Показатели оценки результата**	**Оценка (да / нет)**
ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам ОК 02 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках ПК 3.1 Осуществлять выбор готовых моделей искусственного интеллекта ПК 3.2 Формировать сценарии обучения готовых моделей искусственного интеллекта ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта ПК 3.4 Контролировать результат обучения ПК 3.5 Оформлять результат проведения процедуры обучения ПК 3.6 Формировать запросы для работы с искусственным	Цель, задачи, критерий успеха и выбор подхода логически согласованы.	
	Источник и состав данных описаны; обработаны ошибки/пропуски; исключена утечка.	
	Готовая модель корректно применена; параметры, окружение и порядок запуска задокументированы.	
	Выбраны релевантные метрики, есть сравнение с базовым вариантом, интерпретация и анализ ошибок.	
	Прототип работает на новых входных данных; проверены штатные и ошибочные случаи.	
	Структура соблюдена; выводы доказательны; источники, таблицы, рисунки и код оформлены.	

интеллектом с целью визуализации данных ПК 3.7 Формировать сценарии обучения готовых моделей искусственного интеллекта	Убедительная демонстрация, понимание решений, аргументированные ответы.	

3. Оценка освоения теоретического курса профессионального модуля

3.1. Задания для оценки освоения МДК 03.01

Тема 1.1. Основы искусственного интеллекта и подготовка данных

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Вопросы к устному опросу по теме:

1. Вопрос: Что такое искусственный интеллект и какие задачи он позволяет решать?

Ответ: Искусственный интеллект — это область информатики, занимающаяся разработкой систем, способных выполнять задачи, требующие интеллектуальных способностей человека. Он применяется для распознавания изображений и речи, обработки текста, прогнозирования, анализа данных и автоматизации процессов.

2. Вопрос: Что такое машинное обучение и какие основные типы моделей машинного обучения существуют?

Ответ: Машинное обучение — это направление искусственного интеллекта, позволяющее моделям выявлять закономерности в данных и использовать их для решения задач без явного программирования каждого правила. Основные типы машинного обучения: обучение с учителем, обучение без учителя и обучение с подкреплением.

3. Вопрос: Для чего выполняется предварительная обработка данных перед обучением модели машинного обучения?

Ответ: Предварительная обработка данных выполняется для повышения их качества и пригодности к обучению модели. Она включает удаление дубликатов, обработку пропущенных и

некорректных значений, преобразование признаков и разделение данных на обучающую, валидационную и тестовую выборки.

4. Вопрос: Что такое жизненный цикл модели машинного обучения и какие основные этапы он включает?

Ответ: Жизненный цикл модели машинного обучения — это последовательность этапов разработки и применения модели. Он включает постановку задачи, сбор и подготовку данных, выбор алгоритма, обучение модели, оценку качества, внедрение в практическую деятельность и последующий мониторинг результатов.

5. Вопрос: Какие существуют типы данных для обучения моделей искусственного интеллекта?

Ответ: Для обучения моделей используются числовые, категориальные, текстовые, графические, аудио- и видеоданные. Источниками данных могут быть базы данных, файлы, веб-сайты, датчики, программные интерфейсы API и результаты наблюдений.

6. Вопрос: Чем отличаются обучающая, валидационная и тестовая выборки?

Ответ: Обучающая выборка используется для настройки параметров модели. Валидационная выборка применяется для выбора модели и настройки её гиперпараметров. Тестовая выборка используется для итоговой оценки качества модели на данных, которые не применялись при обучении и настройке.

7. Вопрос: Что такое пропущенные и некорректные значения в наборе данных и как их обрабатывают?

Ответ: Пропущенные значения — это отсутствующие данные в отдельных ячейках набора данных, а некорректные значения — данные, не соответствующие установленным требованиям или допустимому формату. Для их обработки применяют удаление записей, заполнение пропусков средними или медианными значениями, исправление ошибок и преобразование данных к необходимому типу.

8. Вопрос: Для чего проводится визуальный анализ набора данных перед обучением модели?

Ответ: Визуальный анализ позволяет выявить закономерности, распределение значений, выбросы, пропуски и взаимосвязи между признаками. Для анализа используют графики, диаграммы, гистограммы и диаграммы рассеяния. Результаты помогают определить необходимые способы предварительной обработки данных.

9. Вопрос: Что такое признаки и целевая переменная в машинном обучении?

Ответ: Признаки — это характеристики объектов, которые модель использует для построения прогнозов. Целевая переменная — это значение, которое модель должна научиться предсказывать. Например, при прогнозировании стоимости квартиры признаками могут быть площадь, количество комнат и район, а целевой переменной — стоимость квартиры.

10. Вопрос: Почему важно удалять дубликаты и проверять качество данных перед обучением модели?

Ответ: Дубликаты и ошибки могут искажать распределение данных и снижать качество обучения модели. Проверка и очистка набора данных помогают уменьшить количество ошибок, повысить достоверность результатов и улучшить способность модели работать с новыми данными.

Критерии оценки устного опроса:

- оценка «отлично» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
- оценка «хорошо» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
- оценка «удовлетворительно» выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
- оценка «неудовлетворительно» выставляется студенту, если он не смог изложить суть основного вопроса.

Практическая работа №1. Исследование возможностей готовой модели ИИ для решения прикладной задачи

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Задание:

Выберите готовую модель искусственного интеллекта, доступную через веб-сервис или приложение, например ChatGPT, YandexGPT или другую модель. Определите прикладную задачу, которую можно решить с помощью выбранной модели. Проведите несколько экспериментов, используя разные запросы, проанализируйте полученные результаты и оцените качество ответов. Сделайте вывод о возможностях и ограничениях модели, а также определите, насколько эффективно она справляется с поставленной задачей.

Пример решения:

Для выполнения практической работы выбрана модель ChatGPT. В качестве прикладной задачи рассматривается создание программы на Python для вычисления среднего арифметического трёх чисел.

В модель был отправлен запрос: «Напиши программу на Python, которая запрашивает у пользователя три числа и выводит их среднее арифметическое».

Модель предложила следующий код:

```
a = float(input("Введите первое число: "))
b = float(input("Введите второе число: "))
c = float(input("Введите третье число: "))

average = (a + b + c) / 3

print("Среднее арифметическое:", average)
```

Для проверки программы были введены числа 10, 20 и 30. Программа вывела результат 20. Результат соответствует математическому расчёту.

Затем был проведён дополнительный эксперимент: модели предложили объяснить работу программы по строкам. Модель предоставила описание назначения каждой команды, что позволило лучше понять принцип работы кода.

По результатам исследования установлено, что готовая модель искусственного интеллекта может создавать программный код, объяснять его работу и помогать решать учебные задачи. При этом полученные результаты необходимо проверять, поскольку модель может допускать ошибки или предлагать неоптимальные решения.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода — 1 балл.
- Читаемость и стиль кода — 1 балл.
- Применение инструментов разработки и интеграции — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №2.Импорт, первичный анализ и очистка набора данных

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Задание:

Выберите набор данных в формате CSV или Excel, содержащий информацию о некоторой предметной области, например сведения об учениках, товарах, продажах или погоде. Импортируйте набор данных в Python с использованием библиотеки pandas. Выполните первичный анализ данных: определите количество строк и столбцов, изучите названия и типы столбцов, выведите первые записи, проверьте наличие пропущенных значений и дубликатов.

Проведите очистку набора данных: удалите дубликаты, обработайте пропущенные значения и исправьте обнаруженные ошибки. Сохраните очищенный набор данных в отдельный файл и сформулируйте вывод о выполненной работе.

Пример решения (PlantUML):

Выберите готовую модель искусственного интеллекта, доступную через веб-сервис или приложение, например ChatGPT, YandexGPT или другую модель. Определите прикладную задачу, которую можно решить с помощью выбранной модели. Проведите несколько экспериментов, используя разные запросы, проанализируйте полученные результаты и оцените качество ответов. Сделайте вывод о возможностях и ограничениях модели, а также определите, насколько эффективно она справляется с поставленной задачей.

Пример решения

Для выполнения практической работы выбрана модель ChatGPT. В качестве прикладной задачи рассматривается создание программы на Python для вычисления среднего арифметического трёх чисел.

В модель был отправлен запрос: «Напиши программу на Python, которая запрашивает у пользователя три числа и выводит их среднее арифметическое».

Модель предложила следующий код:

```
a = float(input("Введите первое число: "))
b = float(input("Введите второе число: "))
c = float(input("Введите третье число: "))

average = (a + b + c) / 3

print("Среднее арифметическое:", average)
```

Для проверки программы были введены числа 10, 20 и 30. Программа вывела результат 20. Результат соответствует математическому расчёту.

Затем был проведён дополнительный эксперимент: модели предложили объяснить работу программы по строкам. Модель предоставила описание назначения каждой команды, что позволило лучше понять принцип работы кода.

По результатам исследования установлено, что готовая модель искусственного интеллекта может создавать программный код, объяснять его работу и помогать решать учебные задачи. При этом полученные результаты необходимо проверять, поскольку модель может допускать ошибки или предлагать неоптимальные решения.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода — 1 балл.
- Читаемость и стиль кода — 1 балл.
- Применение инструментов разработки и интеграции — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №3. Обработка пропущенных и некорректных значений в данных

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Задание:

Используйте набор данных в формате CSV или Excel, содержащий информацию о некоторой предметной области, например о товарах интернет-магазина, учениках или продажах. Импортируйте данные в Python с использованием библиотеки pandas. Найдите пропущенные значения, некорректные записи и значения, не соответствующие заданным условиям. Выполните обработку обнаруженных проблем: заполните пропущенные значения подходящими данными, исправьте ошибки, приведите значения к необходимым типам и удалите записи, которые невозможно корректно восстановить. После обработки повторно проверьте набор данных и сохраните результат в отдельный файл. Сформулируйте вывод о выполненной работе.

Пример решения:

Для выполнения практической работы используется набор данных о товарах интернет-магазина, содержащий название товара, цену и количество на складе. В исходном файле присутствуют пропущенные значения, отрицательная цена и некорректное количество товара.

Пример содержимого файла products.csv:

```
name,price,quantity
Ноутбук,55000,10
Мышь,1200,25
Клавиатура,,15
Монитор,-5000,8
Наушники,3500,
Колонки,abc,5
```

Для обработки данных используется следующий код на Python:

```
import pandas as pd

# Импорт данных
df = pd.read_csv("products.csv")

# Преобразование столбцов к числовому типу
df["price"] = pd.to_numeric(df["price"], errors="coerce")
df["quantity"] = pd.to_numeric(
    df["quantity"], errors="coerce"
)
```

```
# Замена некорректных значений на пропущенные
df.loc[df["price"] <= 0, "price"] = None
df.loc[df["quantity"] < 0, "quantity"] = None

# Заполнение пропущенных цен медианой
median_price = df["price"].median()
df["price"] = df["price"].fillna(median_price)

# Заполнение пропущенного количества нулём
df["quantity"] = df["quantity"].fillna(0)

# Проверка результата
print("Пропущенные значения:")
print(df.isnull().sum())

print("\nОбработанные данные:")
print(df)

# Сохранение результата
df.to_csv("products_corrected.csv", index=False)
```

В результате обработки текстовое значение abc в столбце цены преобразуется в пропуск, а отрицательная цена товара «Монитор» признаётся некорректной. Пропущенные цены заменяются медианным значением корректных цен, которое составляет 3500. Пропущенное количество товара «Наушники» заменяется нулём, что в данном примере означает отсутствие товара на складе. После обработки данные сохраняются в файл products_corrected.csv.

Вывод: в ходе практической работы были изучены способы выявления и обработки пропущенных и некорректных значений с помощью библиотеки pandas. Выполнено преобразование типов данных, исправление недопустимых значений и заполнение пропусков. Подготовленный набор данных может использоваться для дальнейшего анализа и обучения модели машинного обучения.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода — 1 балл.
- Читаемость и стиль кода — 1 балл.
- Применение инструментов разработки и интеграции — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №4. Подготовка данных для обучения и тестирования модели

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта

ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Задание:

Выберите набор данных в формате CSV или Excel, содержащий информацию о некоторой предметной области, например о стоимости жилья, продажах товаров или успеваемости студентов. Импортируйте набор данных в Python с использованием библиотеки pandas. Проведите предварительный анализ данных, определите признаки и целевую переменную, которые будут использоваться для обучения модели машинного обучения. При необходимости выполните очистку данных и преобразование значений. Разделите набор данных на обучающую и тестовую выборки в соотношении 80:20. Используйте библиотеку scikit-learn для разделения данных. Выведите размеры полученных выборок и сохраните подготовленные данные в отдельные файлы. Сформулируйте вывод о выполненной работе.

Пример решения:

Для выполнения практической работы используется набор данных о квартирах, содержащий информацию об их площади, количестве комнат и стоимости. Необходимо подготовить данные для обучения модели, которая будет прогнозировать стоимость квартиры на основе её характеристик.

Пример содержимого файла apartments.csv:

```
area,rooms,price
35,1,3500000
48,2,4800000
60,2,6000000
75,3,7500000
90,3,9000000
110,4,11000000
```

Для подготовки и разделения данных используется следующий код на Python:

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Импорт набора данных
df = pd.read_csv("apartments.csv")

# Просмотр данных
print(df.head())
print("Размер набора данных:", df.shape)

# Выбор признаков и целевой переменной
X = df[["area", "rooms"]]
y = df["price"]

# Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# Вывод размеров выборок
print("Обучающая выборка:", X_train.shape)
print("Тестовая выборка:", X_test.shape)
```

```
# Сохранение подготовленных данных
X_train.to_csv("X_train.csv", index=False)
X_test.to_csv("X_test.csv", index=False)
y_train.to_csv("y_train.csv", index=False)
y_test.to_csv("y_test.csv", index=False)

print("Подготовка данных завершена.")
```

В данном примере признаки area и rooms используются для прогнозирования стоимости квартиры, которая является целевой переменной price. Параметр test_size=0.2 указывает, что 20% данных выделяется для тестирования, а оставшиеся 80% используются для обучения. Параметр random_state=42 обеспечивает воспроизводимость разделения при повторном запуске программы.

Поскольку исходный набор содержит 6 записей, в обучающую выборку попадут 4 записи, а в тестовую — 2 записи. Полученные выборки сохраняются в отдельные CSV-файлы для дальнейшего использования.

Вывод: в ходе практической работы были освоены выбор признаков и целевой переменной, разделение данных на обучающую и тестовую выборки с помощью библиотеки scikit-learn, а также сохранение подготовленных данных. Выполненная подготовка позволяет использовать набор данных для последующего обучения модели машинного обучения и оценки качества её прогнозов.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода — 1 балл.
- Читаемость и стиль кода — 1 балл.
- Применение инструментов разработки и интеграции — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №5. Формирование и визуальный анализ набора данных для обучения модели

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Задание:

Сформируйте набор данных для обучения модели машинного обучения, выбрав предметную область, например успеваемость студентов, продажи товаров или стоимость жилья. Создайте таблицу с несколькими признаками и целевой переменной и сохраните её в формате CSV.

Импортируйте данные в Python с использованием библиотеки pandas. Выполните визуальный анализ набора данных с помощью библиотек matplotlib и seaborn. Постройте не менее трёх визуализаций: столбчатую диаграмму, гистограмму и диаграмму рассеяния. Проанализируйте распределение значений, выявите возможные закономерности и взаимосвязи между признаками. Сформулируйте вывод о пригодности сформированного набора данных для обучения модели машинного обучения.

Пример решения:

Для выполнения практической работы формируется набор данных об успеваемости студентов. Необходимо исследовать взаимосвязь между количеством часов подготовки к экзамену и полученной оценкой.

Создадим файл students.csv со следующим содержимым:

```
student,hours,grade
Студент 1,1,3
Студент 2,2,3
Студент 3,3,4
Студент 4,4,4
Студент 5,5,5
Студент 6,6,5
Студент 7,2,3
Студент 8,7,5
```

Для визуального анализа используется следующий код на Python:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Импорт набора данных
df = pd.read_csv("students.csv")

# Просмотр данных
print(df.head())
print(df.describe())

# Создание области для графиков
fig, axes = plt.subplots(1, 3, figsize=(15, 4))

# Столбчатая диаграмма оценок
sns.countplot(data=df, x="grade", ax=axes[0])
axes[0].set_title("Распределение оценок")
axes[0].set_xlabel("Оценка")
axes[0].set_ylabel("Количество студентов")

# Гистограмма часов подготовки
axes[1].hist(df["hours"], bins=5, edgecolor="black")
axes[1].set_title("Часы подготовки")
axes[1].set_xlabel("Количество часов")
axes[1].set_ylabel("Количество студентов")

# Диаграмма рассеяния
sns.scatterplot(
    data=df,
```

```
    x="hours",
    y="grade",
    ax=axes[2]
)
axes[2].set_title("Подготовка и оценка")
axes[2].set_xlabel("Часы подготовки")
axes[2].set_ylabel("Оценка")

plt.tight_layout()
plt.show()
```

В результате выполнения программы формируются три графика. Столбчатая диаграмма показывает количество студентов, получивших каждую оценку. Гистограмма отображает распределение времени подготовки к экзамену. Диаграмма рассеяния позволяет исследовать взаимосвязь между количеством часов подготовки и оценками студентов.

По полученным данным можно заметить общую тенденцию: студенты, уделявшие подготовке больше времени, получили более высокие оценки. Однако данный набор данных небольшой и специально сформирован для учебного примера, поэтому выявленная закономерность не доказывает наличие причинно-следственной связи.

Вывод: в ходе практической работы был сформирован набор данных, выполнен его первичный анализ и построены графики с использованием библиотек matplotlib и seaborn. Визуализация позволила изучить распределение признаков и выявить возможную взаимосвязь между ними. Полученный набор данных подходит для демонстрации этапов подготовки данных, однако для построения надёжной модели машинного обучения потребуется более объёмный и репрезентативный набор данных.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода — 1 балл.
- Читаемость и стиль кода — 1 балл.
- Применение инструментов разработки и интеграции — 1 балл.
- Оформление отчёта — 1 балл.

Тема 1.2. Алгоритмы обучения и разработка сценариев обучения готовых моделей

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Вопросы к устному опросу по теме:

1. Вопрос: Какие основные задачи решает машинное обучение?

Ответ: Основными задачами машинного обучения являются классификация, регрессия, кластеризация и выявление закономерностей в данных. Выбор задачи зависит от цели исследования и типа имеющихся данных.

2. Вопрос: Что такое обучение с учителем?

Ответ: Обучение с учителем — это метод машинного обучения, при котором модель обучается на данных, содержащих признаки объектов и правильные ответы. Такой подход применяется для классификации объектов и прогнозирования числовых значений.

3. Вопрос: Что такое обучение без учителя?

Ответ: Обучение без учителя — это метод машинного обучения, при котором модель анализирует данные без заранее заданных правильных ответов. Он применяется для кластеризации, поиска скрытых закономерностей и выявления групп похожих объектов.

4. Вопрос: Чем классификация отличается от регрессии?

Ответ: Классификация предназначена для определения категории или класса объекта, например для распознавания спама. Регрессия используется для прогнозирования числовых значений, например стоимости квартиры или объёма продаж.

5. Вопрос: Что такое кластеризация и для чего она применяется?

Ответ: Кластеризация — это задача машинного обучения, заключающаяся в разделении объектов на группы по сходству их характеристик. Она применяется для сегментации покупателей, группировки документов и анализа поведения пользователей.

6. Вопрос: Что такое гиперпараметры модели машинного обучения?

Ответ: Гиперпараметры — это настройки модели, которые задаются до начала или в процессе организации обучения и влияют на его результат. К ним относятся количество кластеров, глубина дерева решений, скорость обучения и количество итераций. Гиперпараметры настраивают для повышения качества работы модели.

7. Вопрос: Какие способы настройки гиперпараметров существуют?

Ответ: Основными способами настройки гиперпараметров являются ручной подбор, поиск по сетке (Grid Search) и случайный поиск (Random Search). Для сравнения вариантов используют валидационную выборку и метрики качества модели.

8. Вопрос: Что такое сценарий обучения модели машинного обучения?

Ответ: Сценарий обучения — это последовательность действий, необходимых для подготовки данных, выбора модели, настройки параметров, обучения и оценки результатов. Он позволяет организовать процесс обучения и повторять его при изменении данных или настроек.

9. Вопрос: Чем обучение готовой модели отличается от её дообучения?

Ответ: Обучение модели предполагает настройку её параметров на выбранном наборе данных. Дообучение готовой модели заключается в дополнительной настройке уже обученной модели на новых или специализированных данных для адаптации к конкретной задаче.

10. Вопрос: Как оценивается качество обученной модели?

Ответ: Качество модели оценивается с помощью метрик, соответствующих решаемой задаче. Для классификации применяются accuracy, precision, recall и F1-мера, для регрессии — MAE, MSE и R2, а для кластеризации — коэффициент силуэта. Результаты оценки помогают определить, насколько хорошо модель решает поставленную задачу.

Критерии оценки устного опроса:

- оценка «отлично» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
- оценка «хорошо» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
- оценка «удовлетворительно» выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
- оценка «неудовлетворительно» выставляется студенту, если он не смог изложить суть основного вопроса.

Практическая работа №6. Создание сценария обучения готовой модели для задачи классификации

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Задание:

Выберите набор данных для решения задачи классификации, например определения вида

цветка по его характеристикам или определения категории товара. Импортируйте данные в Python с использованием библиотеки pandas. Определите признаки и целевую переменную. Разделите данные на обучающую и тестовую выборки. Используя библиотеку scikit-learn, выберите готовый алгоритм классификации, например дерево решений, и обучите модель на подготовленных данных. Выполните предсказания для тестовой выборки, оцените качество модели с помощью метрик accuracy, precision, recall и F1-мера. Сформулируйте вывод о результатах обучения и качестве классификации.

Пример решения:

Для выполнения практической работы используется встроенный набор данных Iris, содержащий характеристики цветков ириса трёх видов. Необходимо создать модель, которая определяет вид цветка по длине и ширине чашелистика и лепестка.

Для реализации сценария обучения используется следующий код на Python:

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report

# Загрузка готового набора данных
iris = load_iris()
X = iris.data
y = iris.target

# Разделение данных на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42,
    stratify=y
)

# Создание модели классификации
model = DecisionTreeClassifier(random_state=42)

# Обучение модели
model.fit(X_train, y_train)

# Предсказание классов
y_pred = model.predict(X_test)

# Оценка качества модели
accuracy = accuracy_score(y_test, y_pred)
```

```
print("Точность модели:", accuracy)
print("\nОтчёт о классификации:")
print(classification_report(y_test, y_pred, target_names=iris.target_names))
```

В данном примере загружается набор данных Iris, содержащий 150 записей и четыре признака каждого цветка. Целевая переменная представляет собой вид ириса. Данные разделяются на обучающую выборку, составляющую 80%, и тестовую выборку, составляющую 20%.

Для классификации используется алгоритм дерева решений DecisionTreeClassifier. Метод fit() выполняет обучение модели, а метод predict() позволяет определить классы объектов тестовой выборки. Функция accuracy_score() рассчитывает долю правильных предсказаний, а classification_report() выводит значения precision, recall и F1-меры для каждого класса.

При запуске данного сценария точность классификации на тестовой выборке составляет приблизительно 1,0, то есть 100%. Конкретный результат зависит от состава тестовой выборки и настроек модели.

Вывод: в ходе практической работы был разработан сценарий обучения модели классификации с использованием библиотеки scikit-learn. Выполнены загрузка данных, разделение на обучающую и тестовую выборки, обучение дерева решений и оценка качества предсказаний. Полученные результаты демонстрируют возможность применения готовых алгоритмов машинного обучения для автоматического определения категорий объектов. При этом высокая точность на небольшой тестовой выборке не гарантирует аналогичного качества модели на новых данных.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №7. Разработка сценария обучения модели для задачи регрессии

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Задание:

Разработать сценарий обучения модели машинного обучения для решения задачи регрессии. В

качестве исходных данных использовать набор сведений о квартирах, содержащий информацию об их площади, количестве комнат, этаже и стоимости.

Необходимо выполнить следующие действия:

Создать или загрузить набор данных для обучения модели.

Провести первичный анализ исходных данных.

Разделить данные на признаки и целевую переменную.

Разделить набор данных на обучающую и тестовую выборки.

Выбрать алгоритм регрессии и обучить модель.

Получить прогноз стоимости квартир на тестовой выборке.

Оценить качество модели с использованием метрик регрессии.

Сформулировать вывод о результатах обучения.

Пример решения:

В качестве примера разработаем модель, которая прогнозирует стоимость квартиры на основе её характеристик. Для реализации используем язык Python и библиотеки pandas и scikit-learn.

1. Подготовка исходных данных

Для простоты создадим небольшой учебный набор данных непосредственно в программе. В реальных проектах данные могут загружаться из CSV-файлов, баз данных или внешних источников.

```
import pandas as pd

data = {
    "area": [30, 45, 50, 60, 65, 70, 80, 85, 90, 100,
             110, 120, 125, 130, 140, 150],
    "rooms": [1, 2, 2, 2, 3, 3, 3, 3, 3, 4,
              4, 4, 4, 4, 5, 5],
    "floor": [1, 3, 5, 2, 4, 6, 3, 7, 5, 8,
              4, 9, 6, 10, 7, 12],
    "price": [3.0, 4.2, 4.8, 5.5, 6.0, 6.5, 7.4, 7.8,
              8.2, 9.5, 10.2, 11.0, 11.5, 12.0, 13.0, 14.0]
```

```
}

df = pd.DataFrame(data)

print(df.head())
print(df.info())
print(df.describe())
```

В наборе данных используются следующие столбцы:

area — площадь квартиры в квадратных метрах;

rooms — количество комнат;

floor — этаж;

price — стоимость квартиры в условных миллионах рублей.

Метод head() выводит первые пять строк таблицы. Метод info() показывает структуру данных и типы столбцов, а describe() позволяет получить основные статистические характеристики.

2. Разделение данных на признаки и целевую переменную

```
X = df[["area", "rooms", "floor"]]
y = df["price"]

print("Признаки:")
print(X.head())

print("Целевая переменная:")
print(y.head())
```

Переменная X содержит признаки, по которым модель будет прогнозировать стоимость квартиры. Переменная y содержит известные значения стоимости, на основе которых модель будет обучаться.

В задаче регрессии целевая переменная обычно является числовой величиной, которую необходимо предсказать.

3. Разделение данных на обучающую и тестовую выборки

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.25,
    random_state=42
)

print("Размер обучающей выборки:", len(X_train))
print("Размер тестовой выборки:", len(X_test))
```

Функция train_test_split() разделяет исходный набор данных на две части:

Обучающая выборка используется для обучения модели.

Тестовая выборка используется для проверки качества прогнозов на данных, которые модель не использовала при обучении.

Параметр test_size=0.25 означает, что 25% данных выделяется для тестирования, а остальные 75% — для обучения. Параметр random_state=42 обеспечивает воспроизводимость разделения.

4. Создание и обучение модели регрессии

В качестве алгоритма используем линейную регрессию.

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()

model.fit(X_train, y_train)

print("Модель обучена.")
print("Коэффициенты модели:", model.coef_)
print("Свободный коэффициент:", model.intercept_)
```

Линейная регрессия устанавливает зависимость между входными признаками и целевой переменной. В данном примере модель пытается определить, как площадь, количество комнат и этаж связаны со стоимостью квартиры.

Метод fit() выполняет обучение модели на обучающей выборке.

5. Получение прогнозов

```
y_pred = model.predict(X_test)

results = pd.DataFrame({
    "Фактическая стоимость": y_test.values,
    "Прогнозируемая стоимость": y_pred
})

print(results.round(2))
```

Метод predict() рассчитывает прогнозируемую стоимость квартир для тестовой выборки. Созданная таблица позволяет сравнить фактические значения с предсказаниями модели.

6. Оценка качества модели

```
from sklearn.metrics import (
    mean_absolute_error,
    mean_squared_error,
    r2_score
)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5
r2 = r2_score(y_test, y_pred)

print("MAE:", round(mae, 2))
print("MSE:", round(mse, 2))
print("RMSE:", round(rmse, 2))
print("R²:", round(r2, 3))
```

Для оценки модели используются следующие метрики:

Метрика

Назначение

MAE

Средняя абсолютная ошибка прогнозирования.

MSE

Средняя квадратичная ошибка, сильнее учитывающая большие отклонения.

RMSE

Корень из средней квадратичной ошибки; выражается в единицах целевой переменной.

R^2

Показывает, насколько хорошо модель описывает изменчивость целевой переменной.

Чем меньше значения MAE, MSE и RMSE, тем меньше ошибки модели. Значение R

2

, близкое к 1, обычно указывает на хорошее соответствие модели данным. Однако оценивать качество необходимо с учётом размера выборки и особенностей задачи.

Важно учитывать, что созданный набор данных небольшой и учебный. Полученные метрики демонстрируют принцип оценки модели, но не позволяют судить о качестве прогнозирования реальных цен на недвижимость.

7. Прогноз стоимости новой квартиры

После обучения модель можно использовать для прогнозирования стоимости объекта с заданными характеристиками.

```
new_apartment = pd.DataFrame({
    "area": [75],
    "rooms": [3],
    "floor": [5]
})

predicted_price = model.predict(new_apartment)

print(
    "Прогнозируемая стоимость квартиры:",
    round(predicted_price[0], 2),
```

"млн рублей"

)

В примере модель получает сведения о квартире площадью 75 м² с тремя комнатами на пятом этаже и рассчитывает предполагаемую стоимость. Прогноз является оценочным и зависит от качества исходных данных и особенностей обученной модели.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №8. Использование готовой модели для решения задачи кластеризации

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Задание:

Разработать сценарий использования готовой модели машинного обучения для решения задачи кластеризации. В качестве исходных данных использовать набор сведений о покупателях интернет-магазина, включающий информацию о возрасте, годовом доходе и активности покупок.

Пример решения:

В качестве примера выполним сегментацию покупателей интернет-магазина. Алгоритм должен самостоятельно разделить покупателей на группы со схожими характеристиками.

Для реализации используем язык Python и библиотеки pandas, scikit-learn и matplotlib.

1. Подготовка исходных данных

Создадим учебный набор данных, содержащий информацию о покупателях.

```
import pandas as pd

data = {
    "age": [
        20, 22, 25, 27, 30,
        35, 38, 40, 42, 45,
        48, 50, 52, 55, 58
    ],
    "income": [
        25, 28, 30, 32, 35,
        45, 48, 50, 52, 55,
        70, 75, 78, 82, 85
    ],
    "purchases": [
        2, 3, 3, 4, 4,
        5, 6, 5, 7, 6,
        8, 9, 8, 10, 9
    ]
}

df = pd.DataFrame(data)

print("Исходные данные:")
print(df)

print("\nОсновная информация:")
print(df.info())

print("\nСтатистические характеристики:")
print(df.describe())
```

В таблице используются следующие признаки:

age — возраст покупателя;

income — условный годовой доход в тысячах рублей;

purchases — количество покупок за выбранный период.

Метод info() позволяет проверить структуру таблицы, а describe() выводит основные статистические характеристики данных.

2. Подготовка признаков

Для кластеризации выберем три признака: возраст, доход и количество покупок.

```
X = df[["age", "income", "purchases"]]

print("Признаки для кластеризации:")
print(X.head())
```

Алгоритм будет искать покупателей со схожими значениями этих признаков. Поскольку показатели измеряются в разных масштабах, перед кластеризацией необходимо выполнить нормализацию.

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)

print("Данные после масштабирования:")
print(X_scaled[:5])
```

StandardScaler преобразует признаки таким образом, чтобы их значения были сопоставимы по масштабу. Это важно, поскольку алгоритм K-Means использует расстояния между объектами и без масштабирования признак с большими числовыми значениями может оказывать чрезмерное влияние на результат.

3. Использование готового алгоритма кластеризации

Воспользуемся алгоритмом K-Means из библиотеки scikit-learn.

```
from sklearn.cluster import KMeans

model = KMeans(
    n_clusters=3,
    random_state=42,
    n_init=10
)

model.fit(X_scaled)

print("Кластеризация выполнена.")
```

В этом примере мы используем готовую реализацию алгоритма, но обучаем её на собственном наборе данных.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №8. Настройка гиперпараметров готовой модели и сравнение результатов обучения

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта
Задание:

Разработать сценарий настройки гиперпараметров готовой модели машинного обучения и сравнить результаты обучения при использовании различных параметров. В качестве примера использовать задачу классификации видов ирисов на основе характеристик их цветков.

Пример решения:

В качестве примера используем набор данных Iris из библиотеки scikit-learn. Он содержит характеристики цветков ириса: длину и ширину чашелистика, а также длину и ширину лепестка. По этим признакам необходимо определить вид растения.

Для классификации используем алгоритм дерева решений (DecisionTreeClassifier). Его гиперпараметры позволяют регулировать сложность модели.

1. Загрузка и подготовка данных

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

# Загрузка набора данных
iris = load_iris()

X = iris.data
y = iris.target

# Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(
```

```
    X,
    y,
    test_size=0.2,
    random_state=42,
    stratify=y
)

print("Размер обучающей выборки:", len(X_train))
print("Размер тестовой выборки:", len(X_test))
```

В переменной X хранятся признаки цветков, а в y — правильные классы растений.

Функция train_test_split() разделяет данные на две части. Обучающая выборка используется для обучения моделей, а тестовая — для проверки качества их прогнозов на ранее не использованных объектах.

Параметр test_size=0.2 выделяет 20% данных для тестирования. Параметр stratify=y сохраняет пропорции классов в обеих выборках.

2. Обучение моделей с разными гиперпараметрами

Создадим три модели дерева решений. Они будут отличаться параметром max_depth, который ограничивает максимальную глубину дерева.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Модель 1: глубина дерева не ограничена
model_1 = DecisionTreeClassifier(
    max_depth=None,
    random_state=42
)

# Модель 2: максимальная глубина равна 2
model_2 = DecisionTreeClassifier(
    max_depth=2,
    random_state=42
)

# Модель 3: максимальная глубина равна 4
model_3 = DecisionTreeClassifier(
    max_depth=4,
    random_state=42
)
```

```
# Обучение моделей
model_1.fit(X_train, y_train)
model_2.fit(X_train, y_train)
model_3.fit(X_train, y_train)

print("Обучение моделей завершено.")
```

Гиперпараметры — это настройки алгоритма, которые задаются до обучения модели. Они отличаются от параметров, которые модель самостоятельно определяет в процессе обучения.

В данном примере:

max_depth=None — глубина дерева не ограничена этим параметром;

max_depth=2 — дерево может иметь глубину не более двух уровней;

max_depth=4 — дерево может иметь глубину не более четырёх уровней.

Более глубокое дерево способно учитывать сложные зависимости в данных, но может слишком точно подстроиться под обучающую выборку. Слишком простое дерево, наоборот, может не выявить необходимые закономерности.

3. Сравнение качества моделей

Проверим качество каждой модели на обучающей и тестовой выборках.

```
models = {
    "Глубина не ограничена": model_1,
    "Глубина 2": model_2,
    "Глубина 4": model_3
}

results = []

for name, model in models.items():
    # Прогнозы на обучающих данных
    train_pred = model.predict(X_train)

    # Прогнозы на тестовых данных
    test_pred = model.predict(X_test)

    # Расчёт точности
    train_accuracy = accuracy_score(
        y_train,
        train_pred
```

```
    )

    test_accuracy = accuracy_score(
        y_test,
        test_pred
    )

    results.append({
        "Модель": name,
        "Точность на обучении": round(train_accuracy, 3),
        "Точность на тесте": round(test_accuracy, 3)
    })

results_df = pd.DataFrame(results)

print(results_df)
```

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Тема 1.3. Оценка, настройка и повышение качества готовых моделей

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.5. Оформлять результат проведения процедуры обучения

Вопросы к устному опросу по теме:

1. Что такое качество модели машинного обучения?

Качество модели машинного обучения — это степень соответствия её прогнозов реальным значениям или правильным ответам. Качество оценивается с помощью специальных метрик, выбор которых зависит от решаемой задачи. Для классификации используют accuracy, precision, recall и F1-score, а для регрессии — MAE, MSE, RMSE и R

2
.

2. Какие основные подходы применяются для оценки качества модели?

Основные подходы включают проверку модели на тестовой выборке, использование кросс-валидации и сравнение результатов нескольких моделей. Тестовая выборка позволяет оценить способность модели работать с новыми данными, а кросс-валидация помогает получить более устойчивую оценку качества.

3. Что такое accuracy и как она рассчитывается?

Accuracy — это доля объектов, для которых модель правильно определила класс, среди всех проверенных объектов. Метрика рассчитывается как отношение количества правильных предсказаний к общему количеству объектов. Например, если модель правильно классифицировала 90 из 100 объектов, accuracy равна 0,9, или 90%. При несбалансированных классах эта метрика может создавать ошибочное впечатление о качестве модели.

4. Что такое precision и recall?

Precision показывает, какая доля объектов, отнесённых моделью к определённому классу, действительно принадлежит этому классу. Recall показывает, какую долю всех объектов этого класса модель смогла обнаружить. Например, при выявлении мошеннических операций precision помогает оценить, сколько отмеченных операций действительно являются мошенническими, а recall — сколько всех мошеннических операций удалось обнаружить.

5. Что такое F1-score и для чего он используется?

F1-score — это гармоническое среднее между precision и recall. Метрика позволяет оценить баланс между точностью положительных предсказаний и полнотой обнаружения объектов нужного класса. Она особенно полезна при работе с несбалансированными наборами данных, когда важно учитывать обе характеристики.

6. Что такое матрица ошибок классификации?

Матрица ошибок — это таблица, которая сопоставляет реальные классы объектов с классами, предсказанными моделью. В задаче бинарной классификации она включает четыре показателя: TP — правильно обнаруженные положительные объекты, TN — правильно определённые отрицательные объекты, FP — отрицательные объекты, ошибочно отнесённые к положительным, и FN — положительные объекты, которые модель не обнаружила. Матрица помогает понять, какие именно ошибки совершает модель.

7. Что такое ROC-кривая и что показывает AUC?

ROC-кривая показывает зависимость доли истинно положительных результатов от доли ложноположительных результатов при изменении порога классификации. Она позволяет оценить способность модели различать два класса при разных настройках порога. AUC — площадь под ROC-кривой. Чем ближе AUC к 1, тем лучше модель обычно разделяет классы. Значение около 0,5 соответствует качеству случайного ранжирования в типичном случае бинарной классификации.

8. Какие метрики используются для оценки моделей регрессии?

Для оценки регрессии применяются MAE, MSE, RMSE и коэффициент детерминации R^2. MAE показывает среднюю абсолютную величину ошибки, MSE — среднее значение квадратов ошибок, RMSE — квадратный корень из MSE, а R^2 характеризует, насколько хорошо модель объясняет изменчивость целевой переменной. Чем меньше MAE, MSE и RMSE, тем меньше ошибки прогнозирования. Значение R^2, близкое к 1, обычно свидетельствует о хорошем соответствии модели данным.

9. Что такое кросс-валидация?

Кросс-валидация — это способ оценки качества модели, при котором исходные данные несколько раз разделяются на обучающую и проверочную части. Например, при пятиблочной кросс-валидации данные делятся на пять частей: на каждом шаге четыре части используются для обучения, а оставшаяся — для проверки. Затем результаты усредняются. Такой подход помогает оценить устойчивость модели и уменьшить зависимость результата от одного случайного разделения данных.

10. Для чего необходимо сравнивать несколько вариантов моделей?

Сравнение моделей позволяет определить, какой алгоритм лучше решает поставленную задачу. Для этого модели обучают и оценивают на одинаковых данных с использованием подходящих метрик. Например, для классификации можно сравнить accuracy и F1-score, а для регрессии — MAE и RMSE. При этом необходимо учитывать не только качество прогнозов, но и скорость работы, сложность модели и требования к вычислительным ресурсам.

11. Что такое настройка гиперпараметров модели?

Настройка гиперпараметров — это подбор параметров алгоритма, которые задаются до или в процессе организации обучения, но не определяются моделью непосредственно из обучающих примеров. Например, для дерева решений можно настроить максимальную глубину, а для K-Means — количество кластеров. Подбор гиперпараметров выполняют вручную или с помощью инструментов автоматического поиска, например GridSearchCV в библиотеке scikit-learn.

12. Какие методы применяются для оптимизации готовых моделей?

Для повышения качества модели применяют настройку гиперпараметров, подбор признаков, обработку пропущенных и ошибочных значений, масштабирование данных, подбор подходящего алгоритма и увеличение объёма качественных обучающих данных. Также можно использовать регуляризацию и ансамблевые методы. Выбор способа зависит от характера задачи и результатов первоначальной оценки модели.

13. Что такое анализ ошибок модели?

Анализ ошибок модели — это исследование случаев, в которых модель выдала неправильные или неточные прогнозы. Для классификации анализируют матрицу ошибок и примеры неверно определённых классов. Для регрессии изучают разницу между реальными и прогнозируемыми значениями. Такой анализ помогает обнаружить проблемы в исходных данных, недостатки признаков и ограничения выбранного алгоритма.

Критерии оценки устного опроса:

- оценка «отлично» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
- оценка «хорошо» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
- оценка «удовлетворительно» выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
- оценка «неудовлетворительно» выставляется студенту, если он не смог изложить суть основного вопроса.

Практическая работа №9. Расчёт и интерпретация основных метрик качества модели

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.5. Оформлять результат проведения процедуры обучения

Задание:

Разработать сценарий оценки качества модели машинного обучения для решения задачи классификации. Рассчитать основные метрики качества: accuracy, precision, recall и F1-score, проанализировать полученные значения и сформулировать вывод о качестве работы модели.

Пример решения:

1. Подготовка данных

Загрузим готовый набор данных Iris, содержащий информацию о трёх видах ирисов.

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

# Загрузка набора данных
iris = load_iris()
```

```
X = iris.data
y = iris.target

# Разделение данных на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42,
    stratify=y
)

print("Размер обучающей выборки:", len(X_train))
print("Размер тестовой выборки:", len(X_test))
```

Переменная X содержит признаки цветков, а y — номера соответствующих классов. Обучающая выборка используется для обучения модели, а тестовая — для проверки качества её прогнозов.

2. Обучение модели классификации

Для классификации используем алгоритм логистической регрессии.

```
from sklearn.linear_model import LogisticRegression

# Создание модели
model = LogisticRegression(max_iter=200)

# Обучение модели
model.fit(X_train, y_train)

print("Модель обучена.")
```

Метод fit() обучает модель на известных примерах. Параметр max_iter=200 задаёт максимальное количество итераций алгоритма оптимизации.

3. Получение прогнозов модели

```
# Получение прогнозов на тестовой выборке
y_pred = model.predict(X_test)

print("Фактические классы:")
print(y_test)

print("Предсказанные классы:")
print(y_pred)
```

Переменная y_test содержит реальные классы цветков, а y_pred — классы, предсказанные моделью. Их сравнение позволяет определить, сколько объектов классифицировано правильно.

4. Расчёт основных метрик качества

```
from sklearn.metrics import (
    accuracy_score,
    precision_score,
    recall_score,
    f1_score
)

# Расчёт метрик
accuracy = accuracy_score(y_test, y_pred)

precision = precision_score(
    y_test,
    y_pred,
    average="macro"
)

recall = recall_score(
    y_test,
    y_pred,
    average="macro"
)

f1 = f1_score(
    y_test,
    y_pred,
    average="macro"
)

print("Accuracy:", round(accuracy, 3))
print("Precision:", round(precision, 3))
print("Recall:", round(recall, 3))
print("F1-score:", round(f1, 3))
```

Параметр average="macro" означает, что метрика рассчитывается отдельно для каждого класса, а затем результаты усредняются с одинаковым весом. Это позволяет учитывать качество классификации каждого из трёх видов ириса независимо от количества объектов класса.

Высокое значение означает, что модель пропускает меньше объектов нужного класса

Значения метрик находятся в диапазоне от 0 до 1. Чем ближе показатель к 1, тем лучше соответствующая характеристика модели. При этом ни одну метрику не следует рассматривать

отдельно от особенностей задачи.

Для многоклассовой классификации, как в примере с Iris, значения precision, recall и F1-score с параметром average="macro" представляют собой среднее по всем классам.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №10. Построение матрицы ошибок и анализ результатов классификации

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.5. Оформлять результат проведения процедуры обучения

Задание:

Разработать сценарий оценки модели машинного обучения с использованием матрицы ошибок. В качестве примера использовать задачу классификации видов ирисов по характеристикам их цветков.

Пример решения:

Для выполнения работы используем Python и библиотеку scikit-learn. Для визуализации матрицы ошибок применим библиотеки matplotlib и seaborn.

1. Загрузка и подготовка данных

Используем набор данных Iris, содержащий сведения о трёх видах ирисов: Setosa, Versicolor и Virginica.

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

# Загрузка данных
iris = load_iris()

X = iris.data
y = iris.target
```

```
# Разделение на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42,
    stratify=y
)

print("Количество объектов для обучения:", len(X_train))
print("Количество объектов для тестирования:", len(X_test))
```

Переменная X содержит четыре характеристики цветков, а y — правильные классы. Для обучения используется 80% данных, остальные 20% предназначены для проверки модели.

Параметр stratify=y помогает сохранить соотношение классов при разделении данных.

2. Обучение модели классификации

В качестве алгоритма используем дерево решений.

```
from sklearn.tree import DecisionTreeClassifier

# Создание модели
model = DecisionTreeClassifier(
    max_depth=3,
    random_state=42
)

# Обучение модели
model.fit(X_train, y_train)

print("Модель обучена.")
```

Алгоритм дерева решений определяет правила, позволяющие относить объекты к определённым классам. Параметр max_depth=3 ограничивает максимальную глубину дерева тремя уровнями.

3. Получение прогнозов

```
# Прогнозирование классов
y_pred = model.predict(X_test)

print("Фактические классы:")
print(y_test)

print("Предсказанные классы:")
print(y_pred)
```

Массив y_test содержит реальные классы цветков, а y_pred — классы, определённые моделью. Сравнение этих массивов позволяет выявить правильные и ошибочные предсказания.

4. Построение матрицы ошибок

```
from sklearn.metrics import confusion_matrix
```

```
# Расчёт матрицы ошибок
cm = confusion_matrix(
    y_test,
    y_pred,
    labels=[0, 1, 2]
)

print("Матрица ошибок:")
print(cm)
```

Матрица ошибок показывает, сколько объектов каждого реального класса модель отнесла к каждому предсказанному классу.

В библиотеке scikit-learn строки матрицы соответствуют реальным классам, а столбцы — предсказанным.

Для многоклассовой классификации диагональные элементы матрицы показывают количество правильных предсказаний. Элементы вне диагонали показывают ошибки классификации.

5. Визуализация матрицы ошибок

Для более наглядного анализа построим тепловую карту.

```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(7, 5))

sns.heatmap(
    cm,
    annot=True,
    fmt="d",
    cmap="Blues",
    xticklabels=iris.target_names,
    yticklabels=iris.target_names
)

plt.xlabel("Предсказанный класс")
plt.ylabel("Реальный класс")
plt.title("Матрица ошибок классификации")
plt.show()
```

Параметр annot=True отображает числовые значения внутри ячеек, а fmt="d" задаёт их вывод в виде целых чисел. Подписи осей позволяют сопоставить реальные и предсказанные виды ирисов.

При анализе диаграммы необходимо обратить внимание на следующие особенности:

большие значения на главной диагонали означают, что модель правильно распознаёт соответствующие классы;

значения вне главной диагонали указывают на случаи, когда модель перепутала классы;

ошибки между двумя похожими видами могут свидетельствовать о том, что их характеристики трудно различить выбранным алгоритмом.

6. Расчёт количества правильных и ошибочных предсказаний

```
import numpy as np

# Количество правильных предсказаний
correct = np.trace(cm)

# Общее количество объектов
total = np.sum(cm)

# Количество ошибочных предсказаний
errors = total - correct

print("Всего объектов:", total)
print("Правильных предсказаний:", correct)
print("Ошибочных предсказаний:", errors)

print(
    "Доля правильных предсказаний:",
    round(correct / total, 3)
)
```

Функция np.trace(cm) рассчитывает сумму элементов главной диагонали матрицы, то есть количество правильно классифицированных объектов.

Общее количество объектов определяется суммой всех элементов матрицы. Разница между общим количеством и числом правильных предсказаний показывает число ошибок.

Доля правильных предсказаний позволяет оценить общую точность модели.

7. Анализ ошибок по отдельным классам

Чтобы определить, какие виды ирисов модель классифицирует правильно, а какие путает, выведем матрицу в виде таблицы с подписями.

```
import pandas as pd

cm_df = pd.DataFrame(
    cm,
    index=iris.target_names,
    columns=iris.target_names
)

print("Матрица ошибок:")
print(cm_df)
```

Например, значение в строке versicolor и столбце virginica показывает количество цветков вида Versicolor, которые модель ошибочно определила как Virginica.

Анализ таких случаев позволяет выявить классы, между которыми модель чаще всего допускает ошибки, и принять решение о необходимости улучшения признаков, настройки алгоритма или использования другой модели.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №11. Оценка модели с использованием ROC-кривой и различных метрик

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.5. Оформлять результат проведения процедуры обучения

Задание:

Разработать сценарий оценки качества модели машинного обучения для решения задачи бинарной классификации с использованием ROC-кривой и основных метрик качества. В качестве примера рассмотреть определение наличия или отсутствия заболевания по результатам обследования пациента на учебном наборе данных.

Пример решения:

Для выполнения работы используем Python и библиотеку scikit-learn. В качестве данных возьмём встроенный учебный набор Breast Cancer Wisconsin, содержащий числовые характеристики клеточных образцов. Модель будет определять, к какому из двух классов относится образец: доброкачественному или злокачественному.

Набор используется исключительно для обучения и демонстрации методов оценки машинного обучения. Результаты модели не предназначены для постановки медицинских диагнозов.

1. Загрузка и подготовка данных

```
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split

# Загрузка набора данных
data = load_breast_cancer()

X = data.data
```

```
# Положительный класс: злокачественный образец
y = (data.target == 0).astype(int)

# Разделение данных на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.2,
    random_state=42,
    stratify=y
)

print("Количество объектов для обучения:", len(X_train))
print("Количество объектов для тестирования:", len(X_test))
```

Переменная X содержит числовые признаки образцов, а y — целевые классы. Значение 1 обозначает злокачественный образец, а 0 — доброкачественный.

Параметр stratify=y сохраняет приблизительно одинаковое соотношение классов в обучающей и тестовой выборках.

2. Создание и обучение модели

Для решения задачи используем логистическую регрессию. Перед обучением выполним стандартизацию признаков, поскольку они могут иметь разные масштабы значений.

```
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression

# Создание модели с предварительной обработкой данных
model = make_pipeline(
    StandardScaler(),
    LogisticRegression(max_iter=2000)
)

# Обучение модели
model.fit(X_train, y_train)

print("Модель успешно обучена.")
```

StandardScaler приводит признаки к сопоставимому масштабу. LogisticRegression обучается определять вероятность принадлежности объекта к положительному классу.

Использование make_pipeline() объединяет обработку признаков и классификатор в одну последовательность операций.

3. Получение прогнозов и вероятностей

```
# Предсказание классов
y_pred = model.predict(X_test)

# Получение вероятностей для положительного класса
y_prob = model.predict_proba(X_test)[:, 1]
```

```
print("Фактические классы:")
print(y_test[:10])

print("Предсказанные классы:")
print(y_pred[:10])

print("Вероятности положительного класса:")
print(y_prob[:10].round(3))
```

Метод predict() определяет класс объекта с использованием установленного моделью порога классификации. По умолчанию для логистической регрессии этот порог соответствует вероятности 0,5.

Метод predict_proba() возвращает вероятности принадлежности к классам. Выражение [:, 1] выбирает вероятности положительного класса — злокачественного образца.

Для построения ROC-кривой используются именно вероятности или оценки модели, а не только итоговые предсказанные классы.

```
4. Расчёт основных метрик качества
from sklearn.metrics import (
    accuracy_score,
    precision_score,
    recall_score,
    f1_score
)

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print("Accuracy:", round(accuracy, 3))
print("Precision:", round(precision, 3))
print("Recall:", round(recall, 3))
print("F1-score:", round(f1, 3))
```

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №12. Применение кросс-валидации и сравнение нескольких вариантов модели

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке

Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.2. Формировать сценарии обучения готовых моделей искусственного интеллекта
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.5. Оформлять результат проведения процедуры обучения

Задание:

Разработать сценарий оценки и сравнения нескольких моделей машинного обучения с использованием кросс-валидации. В качестве примера использовать задачу классификации видов ирисов по характеристикам их цветков.

Пример решения:

Для выполнения работы используем язык Python и библиотеку scikit-learn. Сравним три алгоритма классификации:

логистическую регрессию;

дерево решений;

метод ближайших соседей.

В качестве исходных данных используем набор Iris, содержащий характеристики цветков трёх видов ирисов.

1. Загрузка и подготовка данных

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

# Загрузка набора данных
iris = load_iris()

X = iris.data
y = iris.target

# Выделение отдельной тестовой выборки
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
```

```
    test_size=0.2,
    random_state=42,
    stratify=y
)

print("Обучающая выборка:", X_train.shape)
print("Тестовая выборка:", X_test.shape)
```

Переменная X содержит признаки цветков, а y — соответствующие классы. В тестовую выборку выделяется 20% исходных данных, а оставшиеся 80% используются для сравнения и выбора модели.

Тестовая выборка не должна участвовать в кросс-валидации, поскольку она предназначена для итоговой независимой оценки выбранного алгоритма.

2. Создание вариантов моделей

Подготовим три алгоритма для сравнения.

```
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier

models = {
    "Логистическая регрессия": make_pipeline(
        StandardScaler(),
        LogisticRegression(max_iter=1000)
    ),

    "Дерево решений": DecisionTreeClassifier(
        max_depth=3,
        random_state=42
    ),

    "Метод ближайших соседей": make_pipeline(
        StandardScaler(),
        KNeighborsClassifier(n_neighbors=5)
```

```
    )
}

print("Подготовлено моделей:", len(models))
```

Логистическая регрессия используется для классификации на основе вероятностной модели. Дерево решений формирует правила классификации, а метод ближайших соседей определяет класс нового объекта на основе классов похожих объектов.

Для логистической регрессии и метода ближайших соседей используется стандартизация признаков, поскольку масштаб характеристик влияет на работу этих алгоритмов. Применение Pipeline гарантирует, что масштабирование выполняется отдельно на обучающей части каждого разбиения, предотвращая утечку данных.

3. Настройка кросс-валидации

Используем пятикратную стратифицированную кросс-валидацию.

```
from sklearn.model_selection import StratifiedKFold, cross_val_score

cv = StratifiedKFold(
    n_splits=5,
    shuffle=True,
    random_state=42
)

print("Количество разбиений:", cv.get_n_splits())
```

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Тема 1.4. Применение и интеграция готовых моделей в информационных системах
Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке

Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.5. Оформлять результат проведения процедуры обучения
ПК 3.6. Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Вопросы к устному опросу по теме:

1. Что такое развертывание серверного программного приложения?

Развертывание — это процесс подготовки и размещения приложения в среде, где оно будет запускаться и обслуживать запросы пользователей или других программ. Он включает установку необходимых компонентов, настройку параметров, установку зависимостей, запуск приложения и проверку его работоспособности.

2. Чем отличается среда разработки от среды эксплуатации?

Среда разработки используется программистом для написания и отладки кода. В ней могут быть включены дополнительные инструменты диагностики и тестирования. Среда эксплуатации — это среда, в которой приложение работает после развертывания и обрабатывает реальные запросы. Она должна быть настроена на стабильную и безопасную работу.

3. Что такое программное окружение серверного приложения?

Программное окружение — это совокупность компонентов, необходимых для запуска приложения: операционная система, интерпретатор или среда выполнения, библиотеки, зависимости, переменные окружения, сетевые настройки и другие параметры.

4. Что такое зависимость программного приложения?

Зависимость — это внешняя библиотека, пакет или программный компонент, который использует приложение для выполнения своих функций. Например, серверное приложение на Python может использовать Flask для создания API, а библиотеку `requests` — для отправки HTTP-запросов.

5. Зачем фиксировать версии зависимостей?

Фиксация версий позволяет обеспечить воспроизводимость установки и запуска приложения. Если использовать разные версии библиотек, программа может работать по-разному или перестать запускаться из-за изменений в их интерфейсах. В Python версии зависимостей можно указывать в файле `requirements.txt`.

6. Что такое виртуальное окружение Python и зачем оно нужно?

Виртуальное окружение — это изолированная среда Python, в которой хранятся зависимости конкретного проекта. Оно позволяет использовать разные версии библиотек в разных проектах, не изменяя глобальное окружение Python. Для создания окружения применяется команда `python -m venv .venv`.

7. Что такое переменные окружения и как они используются на сервере?

Переменные окружения позволяют передавать приложению настройки без изменения исходного кода. Например, с их помощью можно задавать порт сервера, режим работы, адрес базы данных и секретные ключи. Чувствительные значения, такие как пароли, не следует размещать в открытом виде в исходном коде или публиковать в репозитории.

8. Почему серверное приложение необходимо проверять после развертывания?

После развертывания могут возникнуть ошибки конфигурации, отсутствовать зависимости, быть недоступна база данных или неправильно настроены сетевые параметры. Проверка позволяет убедиться, что приложение запускается, принимает запросы и возвращает ожидаемые ответы.

2. Контейнеризация серверных приложений и основы автоматизации развертывания

9. Что такое контейнеризация?

Контейнеризация — это способ упаковки приложения вместе с необходимыми библиотеками, настройками запуска и другими компонентами в изолированную среду — контейнер. Это упрощает перенос приложения между компьютерами и серверами с совместимой контейнерной инфраструктурой.

10. Что такое Docker?

Docker — это платформа для создания, распространения и запуска контейнеров. С её помощью можно собрать образ серверного приложения и запустить на его основе контейнер с изолированным окружением.

11. Чем контейнер отличается от виртуальной машины?

Контейнер обычно использует ядро операционной системы хоста и изолирует процессы приложения. Виртуальная машина запускает отдельную гостевую операционную систему на виртуальном оборудовании. Поэтому контейнеры, как правило, требуют меньше ресурсов и быстрее запускаются, чем полноценные виртуальные машины.

12. Что такое образ контейнера?

Образ контейнера — это подготовленный шаблон, содержащий файловую систему, приложение, необходимые зависимости и инструкции для запуска. На его основе создаются контейнеры. Например, образ может включать Python, Flask, исходный код API и команду запуска сервера.

13. Что такое контейнер?

Контейнер — это запущенный экземпляр образа. В нём выполняется приложение, которое может принимать запросы, обрабатывать данные и взаимодействовать с другими компонентами системы.

14. Для чего предназначен файл `Dockerfile`?

`Dockerfile` содержит инструкции по сборке образа контейнера. В нём можно указать базовый образ, рабочую директорию, копирование файлов проекта, установку зависимостей, открываемый порт и команду запуска приложения.

15. Зачем серверному приложению проброс портов контейнера?

Проброс портов связывает порт внутри контейнера с портом хостовой системы. Благодаря этому приложение, работающее внутри контейнера, становится доступно извне. Например, запись `-p 5000:5000` связывает порт 5000 хоста с портом 5000 контейнера.

16. Что такое автоматизация развертывания?

Автоматизация развертывания — это выполнение повторяющихся операций с помощью скриптов, инструментов сборки и систем автоматизации. Она может включать сборку приложения, запуск тестов, создание образа, публикацию версии и запуск сервиса.

17. Что такое CI/CD?

CI/CD — это подход к автоматизации разработки и выпуска программного обеспечения. CI (непрерывная интеграция) предполагает регулярное объединение изменений кода и автоматические проверки. CD может означать непрерывную доставку подготовленных изменений или непрерывное развертывание их в рабочую среду.

18. Какие преимущества дает контейнеризация серверных ИИ-решений?

Контейнеризация помогает воспроизводить окружение, упрощает установку зависимостей и перенос приложения между серверами. Для ИИ-решений это особенно полезно, поскольку им могут требоваться определённые версии библиотек машинного обучения. При этом для работы с GPU необходимо отдельно учитывать совместимость драйверов, библиотек и настроек контейнера.

Критерии оценки устного опроса:

- оценка «отлично» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
- оценка «хорошо» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
- оценка «удовлетворительно» выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
- оценка «неудовлетворительно» выставляется студенту, если он не смог изложить суть основного вопроса.

Практическая работа №13. Проектирование информационной системы с интеграцией готовой ИИ-модели

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.5. Оформлять результат проведения процедуры обучения
ПК 3.6. Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Задание:

Необходимо разработать проект информационной системы для учебного заведения — «Умный

помощник студента».

Система должна помогать студентам получать ответы на вопросы об учебном процессе. Для формирования ответов используется готовая ИИ-модель, подключаемая через API. Разрабатывать и обучать собственную модель не требуется.

Например, студент может задать вопросы:

«Когда заканчивается первая пара?»

«Какие документы нужны для прохождения практики?»

«Как связаться с учебной частью?»

Информационная система принимает вопрос, передаёт его ИИ-модели, получает ответ и отображает его пользователю.

Необходимо выполнить следующие задания

Определить назначение системы. Сформулировать цель создания системы и перечислить её основные функции.

Определить пользователей. Указать, кто будет работать с системой и какие действия сможет выполнять.

Спроектировать архитектуру. Определить основные компоненты системы и описать их взаимодействие.

Выбрать готовую ИИ-модель. Указать пример подходящей модели и способ её подключения.

Определить входные и выходные данные. Описать, какие данные передаются между пользователем, системой и моделью.

Разработать алгоритм работы. Представить последовательность действий системы при обработке вопроса.

Продумать обработку ошибок. Описать действия системы, если модель недоступна, запрос некорректен или ответ не получен.

Предложить способы проверки системы. Привести не менее трёх тестовых сценариев.

Пример решения:

2.1. Назначение информационной системы

Название: «Умный помощник студента».

Цель: автоматизировать предоставление справочной информации об учебном процессе с помощью готовой ИИ-модели.

Основные функции:

приём вопросов от студентов;

проверка корректности введённого запроса;

передача запроса готовой ИИ-модели;

получение и отображение ответа;

обработка ошибок при обращении к модели;

сохранение истории диалога при наличии соответствующей функции.

2.2. Пользователи системы

Пользователь

Возможности

Студент

Задаёт вопросы, получает ответы

Администратор

Настраивает систему, контролирует доступность сервиса

Сотрудник учебной части

Актуализирует справочную информацию, используемую системой

2.3. Архитектура системы

1. Пользователь

Вводит вопрос и получает ответ

2. Пользовательский интерфейс

Веб-страница с полем ввода и областью ответа

3. Серверная часть

Проверяет запрос, обращается к API и обрабатывает результат

4. Готовая ИИ-модель

Анализирует вопрос и формирует ответ

Ответ возвращается по той же цепочке в обратном направлении.

Дополнительно система может содержать базу данных с расписанием, контактами и правилами учебного заведения. При необходимости сервер передаёт модели актуальные сведения из этой базы, чтобы ответы основывались на проверенных данных.

2.4. Выбор готовой ИИ-модели

В качестве примера можно использовать готовую языковую модель, доступную через API, например модель семейства GPT или другую совместимую модель.

Способ интеграции — API-запрос от серверной части системы.

Почему выбран этот вариант:

не требуется создавать модель с нуля;

не нужно самостоятельно обучать нейронную сеть;

модель можно использовать через программный интерфейс;

при необходимости можно заменить провайдера или модель.

Важно: ключ API хранится на сервере, а не в коде веб-страницы, доступном пользователю.

2.5. Входные и выходные данные

Этап

Входные данные

Выходные данные

Ввод вопроса

Текст студента

Запрос для обработки

Проверка

Полученный текст

Разрешённый запрос или сообщение об ошибке

Обращение к модели

Текст вопроса и контекст

Ответ от ИИ-модели

Обработка ответа

Результат API

Текст ответа или сообщение о сбое

Отображение

Подготовленный ответ

Ответ на экране студента

2.6. Алгоритм работы системы

```
flowchart TD
    A([Начало]) --> B[Студент вводит вопрос]
    B --> C{Вопрос не пустой?}
    C -- Нет --> D[Показать сообщение об ошибке]
    D --> B
    C -- Да --> E[Передать запрос на сервер]
    E --> F[Сформировать запрос к API модели]
    F --> G{Модель ответила?}
    G -- Нет --> H[Показать сообщение о временной ошибке]
    G -- Да --> I[Проверить и обработать ответ]
    I --> J[Показать ответ студенту]
    H --> K([Завершение])
    J --> K
```

2.7. Пример взаимодействия

Вопрос студента:

«Какие документы нужны для прохождения производственной практики?»

Действия системы:

Принимает текст вопроса.

Проверяет, что поле не пустое.

Передаёт запрос на сервер.

Сервер при необходимости получает сведения из базы данных колледжа и формирует запрос к ИИ-модели.

Модель формирует ответ.

Система отображает ответ студенту.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №14. Настройка взаимодействия информационной системы с готовой моделью через программный интерфейс

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.5. Оформлять результат проведения процедуры обучения
ПК 3.6. Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Задание:

В рамках разработки информационной системы «Умный помощник студента» необходимо реализовать взаимодействие серверной части приложения с готовой языковой ИИ-моделью через программный интерфейс (API).

Система должна принимать текстовый вопрос от пользователя, передавать его модели, получать ответ и отображать результат в интерфейсе.

Необходимо выполнить следующие задания:

Изучить назначение API и принцип взаимодействия информационной системы с готовой ИИ-моделью.

Выбрать сервис, предоставляющий доступ к готовой модели через API.

Определить структуру запроса: адрес API, метод HTTP, заголовки и тело запроса.

Реализовать отправку запроса к модели из серверной части приложения.

Получить ответ модели и извлечь из него текстовое содержимое.

Реализовать обработку ошибок: отсутствие соединения, некорректный запрос, ошибка авторизации, превышение времени ожидания.

Проверить работу приложения на нескольких тестовых запросах.

Описать результаты тестирования и сделать вывод.

Важно: если для выбранного сервиса требуется API-ключ, необходимо использовать собственный ключ и хранить его в переменных окружения. Не публикуйте ключ в исходном коде и не

отправляйте его преподавателю.

Пример решения:

```
import logging
from flask import Flask, jsonify, request

app = Flask(__name__)

logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s %(levelname)s %(name)s: %(message)s"
)

logger = logging.getLogger("subjects_api")

subjects = [
    {
        "id": 1,
        "name": "Разработка программных модулей",
        "hours": 120
    },
    {
        "id": 2,
        "name": "Базы данных",
        "hours": 90
    }
]


@app.route("/api/subjects", methods=["GET"])
def get_subjects():
    logger.info("Запрошен список учебных дисциплин")
    return jsonify(subjects), 200


@app.route("/api/subjects/<int:subject_id>", methods=["GET"])
def get_subject(subject_id):
    for subject in subjects:
        if subject["id"] == subject_id:
            logger.info(
                "Найдена учебная дисциплина с id=%s",
                subject_id
            )
            return jsonify(subject), 200

    logger.warning(
        "Учебная дисциплина с id=%s не найдена",
        subject_id
```

```
    )

    return jsonify({
        "error": "Учебная дисциплина не найдена"
    }), 404


@app.route("/api/subjects", methods=["POST"])
def create_subject():
    data = request.get_json(silent=True)

    if not isinstance(data, dict):
        logger.warning("Получен запрос с некорректным JSON")
        return jsonify({
            "error": "Тело запроса должно содержать JSON-объект"
        }), 400

    name = data.get("name")
    hours = data.get("hours")

    if not isinstance(name, str) or not name.strip():
        logger.warning("Не указано корректное название дисциплины")
        return jsonify({
            "error": "Поле name должно быть непустой строкой"
        }), 400

    # bool является подклассом int в Python,
    # поэтому проверяем тип отдельно.
    if type(hours) is not int or hours <= 0:
        logger.warning("Получено некорректное значение hours")
        return jsonify({
            "error": "Поле hours должно быть положительным целым числом"
        }), 400

    new_id = max(
        (subject["id"] for subject in subjects),
        default=0
    ) + 1

    new_subject = {
        "id": new_id,
        "name": name.strip(),
        "hours": hours
    }

    subjects.append(new_subject)

    logger.info(
        "Создана учебная дисциплина с id=%s",
        new_id
    )

    return jsonify(new_subject), 201
```

```
@app.errorhandler(500)
def internal_server_error(error):
    logger.exception("Внутренняя ошибка сервера")

    return jsonify({
        "error": "Внутренняя ошибка сервера"
    }), 500


@app.route("/health", methods=["GET"])
def health():
    return jsonify({"status": "ok"}), 200


if __name__ == "__main__":
    app.run(host="127.0.0.1", port=5000, debug=False)
```

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №15. Разработка сценария применения обученной модели для решения прикладной задачи

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.5. Оформлять результат проведения процедуры обучения
ПК 3.6. Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Задание:

Описать прикладную задачу и объяснить, зачем в ней применяется ИИ-модель.
Определить пользователей системы и их потребности.
Выбрать подходящий тип готовой обученной модели и обосновать выбор.
Определить входные данные, выходные данные и формат результата.
Разработать сценарий применения модели от поступления обращения до получения результата.
Описать действия системы для каждой категории обращения.

Предусмотреть обработку неопределённых или ошибочных результатов.
Подготовить не менее пяти тестовых обращений и указать ожидаемые категории.
Определить критерии оценки качества работы модели.
Результат работы: документ с описанием прикладной задачи, характеристиками модели, сценарием её применения, блок-схемой алгоритма и таблицей тестирования.

Пример решения:
3.1. Описание прикладной задачи
Название системы: «Автоматическая классификация обращений пользователей».
Назначение: автоматическое определение типа текстового обращения для ускорения его обработки.
Основные функции:
получение текста обращения;
проверка корректности входных данных;
передача текста обученной модели;
определение категории обращения;
формирование результата классификации;
передача обращения соответствующему сотруднику;
отправка неоднозначных обращений на ручную проверку.
3.2. Выбор обученной модели
Для решения задачи используется готовая модель классификации текста, предназначенная для определения категории или тональности текстовых сообщений.
Например, можно использовать предварительно обученную модель классификации текста из библиотеки Hugging Face Transformers или готовый облачный сервис классификации.
В данном примере предполагается, что модель настроена на три целевые категории: «Вопрос», «Жалоба» и «Благодарность».
Почему подходит этот тип модели:
входные данные представлены текстом;
результат должен относиться к одной из заранее определённых категорий;
модель позволяет автоматизировать первичную сортировку сообщений;
результат можно использовать в существующей информационной системе.
Важно: обычная модель анализа тональности не обязательно умеет различать вопросы, жалобы и благодарности. Необходимо выбирать модель, которая обучена на соответствующей задаче, либо адаптировать готовую модель.
3.3. Входные и выходные данные
Параметр
Описание
Пример
Входные данные
Текст обращения
«Уже третий день не работает личный кабинет»
Тип модели
Классификация текста
Классификатор обращений
Выходные данные
Категория обращения
Жалоба
Дополнительный результат
Оценка уверенности модели, если поддерживается
0,94
Действие системы
Маршрутизация обращения
Передать ответственному сотруднику

Оценка уверенности зависит от конкретной модели и не всегда является достоверной вероятностью правильного ответа. Порог автоматической обработки необходимо определять экспериментально.
3.4. Сценарий применения модели
Рассмотрим пример: пользователь отправляет сообщение «Не могу войти в личный кабинет, помогите решить проблему».
Шаг 1. Получение обращения
Система принимает текст от пользователя
Шаг 2. Проверка данных
Проверка наличия текста и допустимой длины
Шаг 3. Применение обученной модели
Модель анализирует текст и определяет категорию
Шаг 4. Проверка результата
Проверка категории и, если доступна, оценки уверенности
Шаг 5. Выполнение действия
Передача обращения ответственному сотруднику или на ручную проверку
В рассматриваемом случае модель определяет категорию «Жалоба», после чего информационная система направляет обращение сотруднику технической поддержки.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №16. Разработка и тестирование комплексного сценария обработки данных с использованием ИИ-модели
Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.1. Осуществлять выбор готовых моделей искусственного интеллекта
ПК 3.5. Оформлять результат проведения процедуры обучения
ПК 3.6. Формировать запросы для работы с искусственным интеллектом с целью визуализации данных
ПК 3.7. Формировать сценарии обучения готовых моделей искусственного интеллекта

Задание:

Описать назначение системы и определить её пользователей.
Выбрать готовую ИИ-модель и обосновать её применение.
Разработать последовательность этапов обработки данных.
Определить входные и выходные данные каждого этапа.
Разработать алгоритм взаимодействия компонентов системы.
Реализовать комплексный сценарий обработки отзывов.
Предусмотреть обработку пустых, некорректных и неоднозначных данных.

Провести тестирование минимум на пяти различных отзывах.
Сохранить результаты обработки в JSON-файл.
Проанализировать результаты тестирования и сформулировать вывод.

Пример решения:
Установка библиотеки:
pip install requests
Перед запуском необходимо задать переменные окружения AI_API_KEY, AI_API_URL и AI_MODEL.

```
import os
import json
import requests

API_KEY = os.getenv("AI_API_KEY")
API_URL = os.getenv("AI_API_URL")
MODEL = os.getenv("AI_MODEL")

def analyze_review(review):
    review = review.strip()

    if not review:
        raise ValueError("Отзыв не должен быть пустым")

    headers = {
        "Authorization": f"Bearer {API_KEY}",
        "Content-Type": "application/json"
    }

    prompt = f"""
Проанализируй отзыв студента.

Верни только JSON со следующими полями:
sentiment: positive, negative или neutral;
summary: краткое содержание отзыва;
issues: список конкретных проблем;
needs_review: true или false.

needs_review должно быть true, если в отзыве
указана проблема, требующая проверки сотрудником.

Не придумывай факты, которых нет в отзыве.

Отзыв:
{json.dumps(review, ensure_ascii=False)}
"""

    response = requests.post(
        API_URL,
        headers=headers,
        json={
            "model": MODEL,
            "messages": [
                {
                    "role": "user",
```

```
                "content": prompt
            }
        ],
        "temperature": 0
    },
    timeout=30
)

response.raise_for_status()

data = response.json()
content = data["choices"][0]["message"]["content"]

result = json.loads(content)

if result["sentiment"] not in {
    "positive", "negative", "neutral"
}:
    raise ValueError("Недопустимая тональность")

if not isinstance(result["summary"], str):
    raise ValueError("Некорректное содержание")

if not isinstance(result["issues"], list):
    raise ValueError("Список проблем должен быть списком")

if not isinstance(result["needs_review"], bool):
    raise ValueError("Некорректный флаг проверки")

if not all(
    isinstance(issue, str)
    for issue in result["issues"]
):
    raise ValueError("Некорректный элемент списка проблем")

return result

reviews = [
    "Преподаватель объясняет материал понятно, занятия интересные.",
    "Компьютеры постоянно зависают, невозможно выполнять задания.",
    "Сегодня прошла лекция по базам данных.",
    "Спасибо преподавателю за помощь с проектом!",
    " "
]

results = []

for review_id, review in enumerate(reviews, start=1):
    try:
        analysis = analyze_review(review)

        results.append({
            "id": review_id,
            "review": review,
```

```
            **analysis
        })

    except ValueError as error:
        results.append({
            "id": review_id,
            "review": review,
            "error": str(error)
        })

    except (requests.RequestException, KeyError,
            TypeError, json.JSONDecodeError) as error:
        results.append({
            "id": review_id,
            "review": review,
            "error": f"Ошибка обработки: {error}"
        })

with open("review_results.json", "w", encoding="utf-8") as file:
    json.dump(
        results,
        file,
        ensure_ascii=False,
        indent=4
    )

print("Обработка завершена.")
print("Результаты сохранены в review_results.json")
```

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

3.2. Задания для оценки освоения МДК 03.02

Тема 1.1. Основы интеграции ИИ в информационные системы

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.8. Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов

Вопросы к устному опросу по теме:

Вопрос: Что такое информационная система?

Ответ: Информационная система — это совокупность программ, данных и технических средств для сбора, хранения, обработки и передачи информации.

Вопрос: Какова роль информационных систем в управлении данными?

Ответ: Они обеспечивают хранение, обработку, поиск и передачу данных для решения задач организации.

Вопрос: Какие основные виды информационных систем существуют?

Ответ: Управленческие, бухгалтерские, банковские, образовательные, медицинские, CRM- и ERP-системы.

Вопрос: Что такое архитектура информационной системы?

Ответ: Это структура системы, определяющая её компоненты, их функции и способы взаимодействия.

Вопрос: Какие основные компоненты входят в информационную систему?

Ответ: Пользовательский интерфейс, серверная часть, база данных и программные модули.

Вопрос: Что такое искусственный интеллект?

Ответ: Искусственный интеллект — это технологии, позволяющие компьютерам выполнять задачи, требующие анализа данных, распознавания образов и принятия решений.

Вопрос: Какие основные направления искусственного интеллекта существуют?

Ответ: Машинное обучение, обработка естественного языка, компьютерное зрение, распознавание речи и экспертные системы.

Вопрос: Зачем интегрировать ИИ в информационные системы?

Ответ: Для автоматизации процессов, анализа больших объёмов данных, прогнозирования и поддержки принятия решений.

Вопрос: Где применяется ИИ в информационных системах?

Ответ: В медицине, образовании, банковской сфере, торговле, логистике и управлении предприятиями.

Вопрос: Как ИИ анализирует данные?

Ответ: С помощью алгоритмов и моделей, которые выявляют закономерности, классифицируют данные и формируют прогнозы.

Вопрос: Что такое обучение модели ИИ?

Ответ: Это процесс настройки параметров модели на данных для выполнения определённой задачи.

Вопрос: Как ИИ помогает принимать решения?

Ответ: Анализирует данные, выявляет закономерности и предлагает прогнозы или рекомендации.

Вопрос: Как взаимодействуют ИИ и база данных?

Ответ: Информационная система передаёт данные модели ИИ, получает результат обработки и при необходимости сохраняет его в базе данных.

Вопрос: Зачем нужна проверка данных перед передачей модели?

Ответ: Чтобы исключить некорректные значения, пропуски и данные неподходящего формата.

Вопрос: Как ИИ взаимодействует с пользовательским интерфейсом?

Ответ: Пользователь вводит данные через интерфейс, система передаёт их модели и отображает полученный результат.

Вопрос: Что такое API при интеграции ИИ?

Ответ: API — это программный интерфейс, обеспечивающий обмен данными между компонентами информационной системы и ИИ-моделью.

Вопрос: Для чего используется формат JSON?

Ответ: Для структурированной передачи данных между компонентами системы.

Вопрос: Что выполняется на практическом занятии по проектированию ИС с ИИ?

Ответ: Определяются назначение системы, её компоненты, функции и место применения ИИ.

Вопрос: Что означает построение модели информационной системы с интеграцией ИИ?

Ответ: Создание схемы компонентов системы и связей между ними, включая ИИ-модуль.

Вопрос: Как настроить связь между базой данных и ИИ?

Ответ: Организовать получение данных из базы, их подготовку, передачу модели и обработку результата.

Вопрос: Для чего нужна визуализация взаимодействия элементов ИС с ИИ?

Ответ: Чтобы наглядно показать движение данных и последовательность взаимодействия компонентов системы.

Вопрос: Какие этапы включает интеграция ИИ в информационную систему?

Ответ: Выбор задачи и модели, подготовка данных, настройка взаимодействия, тестирование и оценка результатов.

Вопрос: Зачем тестировать информационную систему с ИИ?

Ответ: Чтобы проверить корректность обмена данными, работу модели и обработку ошибок.

Критерии оценки устного опроса:

- оценка «отлично» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
- оценка «хорошо» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
- оценка «удовлетворительно» выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
- оценка «неудовлетворительно» выставляется студенту, если он не смог изложить суть основного вопроса.

Практическое занятие №1. Проектирование информационной системы с ИИ

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.8. Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов

Задание:

Спроектировать информационную систему «Умный помощник студента» с интеграцией ИИ для автоматического ответа на вопросы студентов. Определить основные компоненты системы и порядок их взаимодействия.

Пример решения:

Назначение: автоматические ответы на вопросы студентов об учебном процессе.

Компоненты: пользовательский интерфейс, сервер, ИИ-модель, база данных.

Алгоритм работы: студент вводит вопрос → сервер передаёт его ИИ-модели → модель формирует ответ → сервер возвращает ответ студенту.

Хранение данных: в базе данных сохраняются вопросы и ответы.

Схема системы:

Пользовательский интерфейс

Серверная часть

ИИ-модель

База данных

Вывод: спроектирована информационная система, в которой ИИ-модель отвечает на вопросы студентов, а

сервер обеспечивает взаимодействие компонентов и обмен данными.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода — 1 балл.
- Читаемость и стиль кода — 1 балл.
- Применение инструментов разработки и интеграции — 1 балл.
- Оформление отчёта — 1 балл.

Практическое занятие №2. Построение модели ИС с интеграцией ИИ

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.8. Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов

Задание:

Разработать модель информационной системы «Умный помощник студента» с интеграцией искусственного интеллекта. Определить основные компоненты системы, описать их назначение и построить схему взаимодействия между ними. Указать, какие данные передаются между компонентами и как формируется итоговый результат.

Пример решения:

1. Назначение системы

«Умный помощник студента» предназначен для автоматической обработки вопросов студентов об учебном процессе, расписании, экзаменах и выполнении заданий. ИИ-модель анализирует текст вопроса и формирует ответ.

2. Основные компоненты системы

Пользовательский интерфейс — позволяет студенту вводить вопросы и получать ответы.

Серверная часть — принимает запросы, проверяет данные и организует взаимодействие компонентов.

ИИ-модель — анализирует вопрос и генерирует ответ на основе доступной информации.

База данных — хранит сведения о расписании, учебных дисциплинах, а также историю запросов и ответов.

3. Схема взаимодействия компонентов

Студент

Вводит вопрос

Пользовательский интерфейс

Передаёт запрос серверу

Серверная часть

Проверяет запрос и получает необходимые данные

ИИ-модель

Анализирует вопрос и формирует ответ

База данных

Предоставляет сведения и сохраняет историю

Сервер объединяет результаты работы компонентов и возвращает ответ студенту.

4. Алгоритм работы системы

Студент вводит вопрос, например: «Когда состоится экзамен по программированию?»

Интерфейс передаёт текст вопроса серверу.

Сервер проверяет корректность запроса и при необходимости получает расписание из базы данных.

ИИ-модель анализирует вопрос и формирует ответ с учётом полученной информации.

Сервер передаёт ответ пользовательскому интерфейсу и сохраняет историю обращения.

Студент видит ответ на экране.

5. Пример входных и выходных данных

Элемент

Пример

Входные данные

«Когда экзамен по программированию?»

Данные из базы

Дата экзамена — 20 декабря

Обработка ИИ

Определение смысла вопроса и формирование ответа

Выходные данные

«Экзамен по программированию состоится 20 декабря»

6. Вывод

В ходе практической работы построена модель информационной системы с интеграцией ИИ. Определены основные компоненты, их функции и порядок взаимодействия. Установлено, что база данных предоставляет фактическую информацию, ИИ-модель обрабатывает запрос, а сервер обеспечивает обмен данными между компонентами системы.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода — 1 балл.
- Читаемость и стиль кода — 1 балл.
- Применение инструментов разработки и интеграции — 1 балл.
- Оформление отчёта — 1 балл.

Практическое занятие №3. Настройка связей между базой данных и ИИ в ИС

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.8. Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов

Задание:

Организовать взаимодействие базы данных и ИИ-модели в информационной системе «Умный помощник студента». Создать таблицу с информацией об учебных дисциплинах и расписании, реализовать

получение данных из базы и передачу их ИИ-модели для формирования ответа на вопрос пользователя.

Пример решения:

1. Назначение

Система должна отвечать на вопросы студентов, используя актуальную информацию из базы данных. Например, при вопросе «Когда экзамен по программированию?» система получает дату экзамена из базы

и передаёт её ИИ-модели для формирования понятного ответа.

2. Структура базы данных

Создадим таблицу `exams`, в которой хранятся сведения об экзаменах.

id	subject	exam_date	room
1	Программирование	2026-12-20	305
2	Базы данных	2026-12-22	210
3	Математика	2026-12-25	108

3. Пример получения данных из базы

Для примера используем SQLite — простую базу данных, не требующую отдельного сервера.

```
import sqlite3

connection = sqlite3.connect("college.db")
cursor = connection.cursor()

cursor.execute("""
CREATE TABLE IF NOT EXISTS exams (
    id INTEGER PRIMARY KEY,
    subject TEXT,
    exam_date TEXT,
    room TEXT
)
""")

cursor.execute("""
INSERT OR IGNORE INTO exams
VALUES (1, 'Программирование', '2026-12-20', '305')
""")

cursor.execute(
    "SELECT subject, exam_date, room FROM exams WHERE subject = ?",
    ("Программирование",)
```

```
)

exam = cursor.fetchone()

if exam:
    print("Дисциплина:", exam[0])
    print("Дата экзамена:", exam[1])
    print("Аудитория:", exam[2])

connection.close()
```

Результат выполнения:

```
Дисциплина: Программирование
Дата экзамена: 2026-12-20
Аудитория: 305
```

4. Передача данных ИИ-модели

После получения данных программа формирует контекст для ИИ-модели:

```
question = "Когда экзамен по программированию?"

context = """
Дисциплина: Программирование.
Дата экзамена: 20 декабря 2026 года.
Аудитория: 305.
"""

prompt = f"""
Ответь на вопрос студента только на основе
предоставленных данных.

Данные:
{context}

Вопрос: {question}
"""
print(prompt)
```

Переменную `prompt` можно передать готовой языковой модели через её API. В этом примере запрос к

модели не выполняется: показана подготовка данных для передачи.

5. Схема взаимодействия

Вопрос студента

Программный модуль

Обрабатывает вопрос и выполняет SQL-запрос

База данных SQLite

Возвращает дату и аудиторию экзамена

ИИ-модель

Формирует понятный ответ на основе полученных данных

Ответ студенту

«Экзамен состоится 20 декабря 2026 года, аудитория 305»

6. Вывод

В ходе практической работы создана таблица базы данных, реализовано получение информации с помощью SQL-запроса и подготовлен контекст для передачи ИИ-модели. Настроенная схема позволяет использовать реальные данные информационной системы при формировании ответов. Для полноценной работы необходимо дополнительно подключить API выбранной ИИ-модели и обработать возможные ошибки доступа к базе данных и модели.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода — 1 балл.
- Читаемость и стиль кода — 1 балл.
- Применение инструментов разработки и интеграции — 1 балл.
- Оформление отчёта — 1 балл.

Практическое занятие №4. Визуализация взаимодействия элементов ИС с ИИ

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.8. Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов

Задание:

Разработать визуальную схему взаимодействия компонентов информационной системы «Умный помощник студента» с ИИ-моделью. Отобразить путь передачи данных от пользователя к ИИ, взаимодействие с базой данных и возврат результата. Для каждого компонента указать его назначение.

Пример решения:

1. Назначение системы
«Умный помощник студента» предназначен для автоматического ответа на вопросы об учебном процессе. Система использует базу данных с расписанием и сведениями о дисциплинах, а ИИ-модель формирует понятные ответы на запросы студентов.
2. Основные элементы системы
Пользователь — вводит вопрос.
Пользовательский интерфейс — принимает вопрос и отображает ответ.
Сервер — обрабатывает запрос и управляет взаимодействием компонентов.
База данных — предоставляет актуальные сведения об учебном процессе.
ИИ-модель — анализирует вопрос и формирует ответ.
3. Визуальная схема взаимодействия
Пользователь
Вводит вопрос об экзамене
Пользовательский интерфейс
Передаёт вопрос серверу
Серверная часть
Обрабатывает запрос и организует обмен данными
База данных
Расписание и даты экзаменов
ИИ-модель
Анализирует вопрос и формирует ответ
Сервер получает данные из базы, передаёт необходимый контекст модели и возвращает результат интерфейсу.
4. Последовательность взаимодействия
Студент вводит вопрос: «Когда экзамен по программированию?»
Интерфейс отправляет вопрос серверу.
Сервер определяет, какие сведения необходимы, и выполняет запрос к базе данных.
База данных возвращает дату экзамена и номер аудитории.
Сервер передаёт вопрос и найденные сведения ИИ-модели.
ИИ-модель формирует ответ на основе полученных данных.
Сервер возвращает ответ в интерфейс, и студент видит результат.
5. Пример входных и выходных данных
Этап
Данные
Входной запрос
«Когда экзамен по программированию?»
Ответ базы данных
20 декабря 2026 года, аудитория 305
Результат обработки ИИ
Формирование понятного текстового ответа
Итоговый ответ
«Экзамен по программированию состоится 20 декабря 2026 года в аудитории 305»

6. Вывод
В ходе практической работы построена визуальная модель взаимодействия элементов информационной системы с ИИ. Определены функции компонентов и последовательность обмена данными. Визуализация помогает понять, как пользовательский интерфейс, сервер, база данных и ИИ-модель совместно обеспечивают решение прикладной задачи.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода — 1 балл.
- Читаемость и стиль кода — 1 балл.
- Применение инструментов разработки и интеграции — 1 балл.
- Оформление отчёта — 1 балл.

Тема 1.2. Интеграция ИИ в бизнес-процессы и автоматизация

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.8. Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов

Вопросы к устному опросу по теме:

Вопрос: Что такое бизнес-процесс?

Ответ: Бизнес-процесс — это последовательность действий, направленных на достижение определённого результата в организации.

Вопрос: Какова роль ИИ в автоматизации бизнес-процессов?

Ответ: ИИ помогает автоматизировать повторяющиеся операции, анализировать данные, прогнозировать результаты и принимать решения.

Вопрос: Какие бизнес-процессы можно автоматизировать с помощью ИИ?

Ответ: Обработку заявок, работу с клиентами, составление отчётов, прогнозирование продаж, управление запасами и обработку документов.

Вопрос: Какие бизнес-процессы подходят для автоматизации в первую очередь?

Ответ: Регулярные, трудоёмкие и основанные на большом количестве данных процессы, в которых можно измерить результат автоматизации.

Вопрос: Что такое CRM-система?

Ответ: CRM — это система управления взаимоотношениями с клиентами, которая хранит сведения о клиентах, обращениях и продажах.

Вопрос: Как ИИ применяется в CRM-системах?

Ответ: Для классификации обращений, прогнозирования продаж, анализа потребностей клиентов и формирования персональных рекомендаций.

Вопрос: Что такое ERP-система?

Ответ: ERP — это система управления ресурсами предприятия, объединяющая данные о финансах, персонале, закупках, производстве и запасах.

Вопрос: Как ИИ применяется в ERP-системах?

Ответ: Для прогнозирования потребности в материалах, оптимизации запасов, выявления отклонений и планирования ресурсов.

Вопрос: Что такое бизнес-аналитика на основе ИИ?

Ответ: Это использование ИИ для анализа бизнес-данных, выявления закономерностей и подготовки прогнозов для принятия решений.

Вопрос: Что такое прогнозирование в бизнесе с помощью ИИ?

Ответ: Это определение вероятных будущих показателей на основе исторических данных, например объёма продаж или спроса.

Вопрос: Какие данные используются для прогнозирования продаж?

Ответ: История продаж, цены, сезонность, рекламные кампании, сведения о клиентах и рыночные тенденции.

Вопрос: Что такое оптимизация бизнес-процесса?

Ответ: Это изменение процесса для сокращения затрат и времени выполнения, уменьшения ошибок и повышения производительности.

Вопрос: Как ИИ помогает оптимизировать бизнес-процессы?

Ответ: Выявляет узкие места, прогнозирует загрузку ресурсов, предлагает оптимальные решения и

автоматизирует отдельные этапы работы.

Вопрос: Какие инструменты можно использовать для автоматизации отчётности с ИИ?

Ответ: Электронные таблицы, системы бизнес-аналитики, Python-библиотеки и языковые модели для анализа данных и подготовки текстовых выводов.

Вопрос: Как ИИ помогает создавать отчёты?

Ответ: Анализирует показатели, выявляет изменения, формирует краткие выводы и генерирует текстовые пояснения к результатам.

Вопрос: Как ИИ применяется в управлении проектами?

Ответ: Для оценки сроков, распределения задач, прогнозирования рисков, анализа загрузки сотрудников и подготовки отчётов о ходе проекта.

Вопрос: Что такое автоматизация бизнес-процесса с использованием ИИ?

Ответ: Это организация выполнения одного или нескольких этапов процесса с помощью ИИ без постоянного ручного вмешательства.

Вопрос: Какие этапы включает внедрение ИИ в бизнес-процесс?

Ответ: Анализ процесса, определение задачи, подготовка данных, выбор модели, интеграция, тестирование и оценка эффективности.

Вопрос: Как оценить эффективность внедрения ИИ?

Ответ: Сравнить показатели до и после внедрения: затраты, время выполнения операций, количество ошибок, производительность и качество результата.

Вопрос: Что такое ROI?

Ответ: ROI — показатель окупаемости инвестиций, который помогает оценить экономическую отдачу от внедрения ИИ.

Вопрос: Какие риски возникают при внедрении ИИ в бизнес?

Ответ: Ошибки прогнозирования, утечка данных, предвзятость модели, дополнительные расходы и зависимость от качества исходных данных.

Вопрос: Для чего проводится анализ бизнес-процессов перед внедрением ИИ?

Ответ: Чтобы определить проблемные этапы, оценить возможности автоматизации и выбрать процессы, в которых ИИ принесёт наибольшую пользу.

Вопрос: Что выполняется в практической работе по моделированию бизнес-процесса с ИИ?

Ответ: Разрабатывается схема процесса, определяются точки применения ИИ и описывается взаимодействие компонентов системы.

Вопрос: В чём заключается оптимизация существующего бизнес-процесса с ИИ?

Ответ: В анализе текущего процесса, выявлении недостатков и разработке улучшенного варианта с применением ИИ.

Вопрос: Как выполняется разработка автоматизированного отчёта с ИИ?

Ответ: Подготавливаются данные, выполняется их анализ, формируются показатели и создаются текстовые выводы или визуализации.

Вопрос: Как интегрировать ИИ в систему управления проектами?

Ответ: Подключить ИИ-модель к данным о задачах, сроках и ресурсах, чтобы автоматизировать анализ, прогнозирование и подготовку рекомендаций.

Вопрос: Почему результаты работы ИИ необходимо проверять?

Ответ: Потому что модель может ошибаться, использовать неполные данные или формировать неточные прогнозы. Проверка помогает снизить риски и повысить надёжность решений.

Критерии оценки устного опроса:

- оценка «отлично» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
- оценка «хорошо» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
- оценка «удовлетворительно» выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;

– оценка «неудовлетворительно» выставляется студенту, если он не смог изложить суть основного вопроса.

Практическая работа №5. Анализ бизнес-процессов для внедрения ИИ

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.8. Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов

Задание:

Выбрать бизнес-процесс организации, проанализировать его этапы и определить, какие из них можно автоматизировать с помощью искусственного интеллекта. Описать существующие проблемы, предложить решение с использованием ИИ и определить ожидаемый эффект от внедрения.

Пример решения:

1. Выбранный бизнес-процесс

Обработка обращений клиентов в интернет-магазине.

2. Описание текущего процесса

Клиент отправляет вопрос о заказе, доставке или возврате товара. Сотрудник читает обращение, определяет его тему, ищет необходимую информацию и составляет ответ. При большом количестве обращений время ожидания увеличивается.

3. Анализ проблем

Большое количество однотипных обращений.

Значительные затраты времени сотрудников.

Возможность ошибок при определении темы обращения.

Длительное ожидание ответа клиентом.

4. Возможности внедрения ИИ

Предлагается внедрить ИИ-модель для автоматической классификации обращений и подготовки ответов на типовые вопросы.

Этап процесса

Решение с ИИ

Получение обращения

Автоматический приём текста

Определение темы

Классификация обращения моделью

Поиск информации

Получение сведений из базы данных

Подготовка ответа

Генерация ответа ИИ

Проверка результата

Передача сложных обращений сотруднику

5. Схема процесса с ИИ

Клиент отправляет обращение

Система принимает обращение

ИИ определяет категорию и готовит ответ

Проверка результата

Типовой вопрос

Ответ отправляется клиенту

Сложный вопрос

Передаётся сотруднику

6. Ожидаемый эффект

Сокращение времени обработки типовых обращений.

Снижение нагрузки на сотрудников.

Ускорение обслуживания клиентов.

Более быстрое распределение обращений по категориям.

Для оценки эффективности необходимо сравнить среднее время обработки, количество ошибок и долю автоматически обработанных обращений до и после внедрения ИИ.

7. Вывод

В ходе практической работы проанализирован процесс обработки обращений клиентов, выявлены его недостатки и предложен вариант автоматизации с использованием ИИ. Внедрение модели позволит ускорить обработку типовых запросов и направлять сложные обращения сотрудникам.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №6. Моделирование бизнес-процесса с ИИ

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.8. Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов

Задание:

Разработать модель бизнес-процесса организации с использованием искусственного интеллекта. Выбрать процесс, определить его участников и последовательность действий, указать этап внедрения ИИ и построить схему взаимодействия компонентов. Сравнить процесс до и после внедрения ИИ.

Пример решения:

1. Выбранный бизнес-процесс

Автоматизация обработки заказов в интернет-магазине.

Цель — ускорить обработку заказов, уменьшить количество ошибок и автоматизировать проверку наличия товаров.

2. Описание процесса без ИИ

Клиент оформляет заказ.

Сотрудник проверяет заказ и наличие товара.

Сотрудник уточняет информацию при необходимости.

Заказ подтверждается.

Клиент получает уведомление.

Недостатки: ручная проверка занимает время, возможны ошибки, при большом количестве заказов увеличивается нагрузка на сотрудников.

3. Модель бизнес-процесса с ИИ

В систему интегрируется ИИ-модель, которая анализирует данные заказа, определяет вероятность ошибки и помогает прогнозировать спрос на товары.

4. Схема бизнес-процесса

Клиент оформляет заказ

Система проверяет данные заказа

ИИ анализирует заказ и выявляет возможные ошибки

Проверка наличия товара в базе данных

Ошибок нет

Заказ подтверждается

Есть проблемы

Заказ проверяет сотрудник

Клиент получает уведомление

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №7. Оптимизация существующего бизнес-процесса с ИИ

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.8. Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов

Задание:

Выбрать существующий бизнес-процесс организации, выявить его недостатки и предложить способы оптимизации с использованием искусственного интеллекта. Сравнить процесс до и после внедрения ИИ, определить ожидаемые преимущества и показатели эффективности.

Пример решения:

1. Выбранный бизнес-процесс

Обработка заявок в службе технической поддержки.

Цель оптимизации — сократить время обработки заявок и быстрее направлять обращения нужным специалистам.

2. Анализ существующего процесса

В текущем процессе сотрудник вручную читает заявку, определяет её категорию, устанавливает приоритет и назначает ответственного специалиста.

Основные проблемы:

длительная обработка большого количества заявок;

ошибки при определении категории;

неравномерное распределение нагрузки между сотрудниками;

задержки при решении срочных проблем.

3. Предлагаемая оптимизация с ИИ

В систему внедряется ИИ-модель, которая анализирует текст заявки, определяет её категорию и предлагает приоритет обработки. Затем система назначает ответственного специалиста по заданным правилам.

Этап

До оптимизации

После внедрения ИИ

Анализ заявки

Вручную

Автоматическая классификация

Определение приоритета

Сотрудником

Рекомендация ИИ

Назначение специалиста

Вручную

По категории и загрузке

Обработка типовых заявок

Вручную

Частично автоматизирована

Контроль результата

Вручную

Автоматическое отслеживание статуса

4. Схема оптимизированного процесса

Поступление заявки

ИИ анализирует текст заявки

Определение категории и приоритета

Назначение ответственного специалиста

Решение проблемы и закрытие заявки

5. Оценка эффективности

Для оценки результата сравним условные показатели до и после внедрения ИИ.

Показатель

До внедрения

После внедрения

Среднее время распределения заявки

10 минут

3 минуты
Количество заявок, обработанных за день
60
90
Доля ошибок классификации
12%
5%

Значения приведены для примера. В реальном проекте их необходимо получить по результатам измерений.
Сокращение времени распределения заявки:
\frac{10-3}{10}\times100\%=70\%
Таким образом, при указанных условных значениях время распределения заявки сокращается на 70%.
6. Ожидаемый результат
Ускорение обработки поступающих заявок.
Снижение количества ошибок при классификации.
Более равномерное распределение нагрузки.
Повышение производительности службы поддержки.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №8.Разработка автоматизированных отчётов с ИИ

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.8. Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов

Задание:

Изучить исходные данные о работе организации.

Определить ключевые показатели для отчёта.

Разработать алгоритм автоматического сбора и обработки данных с помощью ИИ.

Сформировать пример отчёта с выводами и рекомендациями.

Оценить преимущества автоматизации отчётности.

Пример решения:

Определите, какие дополнительные показатели можно рассчитать на основе исходных данных.

Пример расчётов:
1. Средний чек
Средний чек показывает, сколько в среднем приносит один заказ.
\text{Средний чек}=\frac{\text{Выручка}}{\text{Количество заказов}}
Для сентября:
\frac{1\,500\,000}{1\,000}=1\,500\text{ руб.}
2. Доля возвратов
Показывает, какая часть заказов была возвращена.
\text{Доля возвратов}=
\frac{\text{Возвраты}}{\text{Заказы}}\times100\%
Для сентября:
\frac{70}{1000}\times100\%=7\%
3. Изменение выручки
Показывает, насколько выручка текущего месяца изменилась по сравнению с предыдущим.
\text{Изменение, \%}=
\frac{\text{Текущая выручка}-\text{Предыдущая выручка}}
{\text{Предыдущая выручка}}\times100\%
Для сентября относительно августа:
\frac{1\,500\,000-1\,380\,000}{1\,380\,000}
\times100\%\approx8{,}7\%
Задание 2. Анализ результатов
На основе исходных данных и расчётов заполните таблицу.
Показатель
Июль
Август
Сентябрь
Средний чек, руб.
1 500
1 500
1 500
Доля возвратов, %
5%
5%
7%
Изменение выручки к предыдущему месяцу
—
15%
8,7%

Проанализируйте полученные результаты. Обратите внимание, что количество заказов и выручка растут, однако доля возвратов в сентябре увеличилась.
Задание 3. Разработка алгоритма автоматизированного отчёта
Опишите последовательность действий системы, которая будет формировать отчёт без необходимости каждый раз выполнять расчёты вручную.
Пример алгоритма:
1. Сбор данных
Получение заказов, выручки и возвратов из информационной системы
2. Проверка данных
Поиск пропусков, дубликатов и некорректных значений

3. Расчёт показателей
Вычисление среднего чека, доли возвратов и динамики выручки
4. Анализ с помощью ИИ
Поиск отклонений, описание тенденций и подготовка рекомендаций
5. Формирование отчёта
Создание таблиц, диаграмм и текстового заключения для руководителя
Задание 4. Формирование отчёта с помощью ИИ
Используйте любой доступный инструмент ИИ для анализа данных. Передайте ему исходную таблицу и запросите краткий аналитический отчёт.
Пример запроса:
Проанализируй данные интернет-магазина за июль, август и сентябрь.
Сравни количество заказов и выручку по месяцам.
Рассчитай динамику выручки в процентах.
Определи долю возвратов в каждом месяце.
Найди показатели, требующие внимания руководителя.
Сформулируй три рекомендации по улучшению работы магазина.
Используй только предоставленные данные. Не придумывай причины изменений, если их нельзя подтвердить исходной таблицей. Отделяй факты от предположений.
Скопировать запрос для ИИ
После получения ответа проверьте правильность числовых расчётов и оцените, соответствуют ли рекомендации данным.
Задание 5. Оценка эффективности автоматизации
Заполните таблицу, сравнив ручное составление отчёта с автоматизированным.
Критерий
Ручной отчёт
Автоматизированный отчёт с ИИ
Сбор данных
Сотрудник переносит данные вручную
Данные загружаются автоматически при настроенной интеграции
Расчёты
Выполняются сотрудником
Рассчитываются по заданным формулам
Анализ тенденций
Требует самостоятельного анализа
ИИ помогает находить закономерности и отклонения
Подготовка выводов
Пишутся вручную
ИИ формирует черновик выводов
Контроль ошибок
Ручная проверка
Автоматические проверки и проверка сотрудником
Обновление
Нужно повторять вручную
Может выполняться по расписанию

Ответьте письменно: какие этапы можно полностью автоматизировать, а какие требуют проверки человека?

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №9. Интеграция ИИ в систему управления проектами
Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.8. Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов

Задание:

Задание 1. Анализ процессов управления проектом

Заполните таблицу, определив, какие процессы можно улучшить с помощью ИИ.

Процесс	**Проблема**	**Возможность применения ИИ**
Планирование	Сложно определить последовательность работ	Предложение этапов и зависимостей между задачами
Распределение задач	Не учитывается текущая загрузка команды	Рекомендации по назначению исполнителей
Контроль сроков	Риск задержки обнаруживается поздно	Анализ статусов и предупреждение о возможных просрочках
Управление рисками	Проблемы выявляются после их возникновения	Поиск потенциальных рисков и предложение мер
Подготовка отчётов	Сотрудник вручную собирает сведения	Автоматическое формирование сводки о ходе проекта

После заполнения таблицы выберите не менее трёх процессов, для которых внедрение ИИ будет наиболее полезным, и объясните свой выбор.

Задание 2. Разработка плана проекта с использованием ИИ

Представьте, что необходимо создать сайт колледжа за 20 рабочих дней. С помощью ИИ нужно сформировать предварительный план работ.

Пример плана:

№	Задача	Длительность	Зависимость	Возможная функция ИИ
1	Сбор требований	2 дня	Нет	Систематизация требований
2	Проектирование структуры сайта	3 дня	Задача 1	Предложение структуры страниц
3	Разработка дизайна	4 дня	Задача 2	Генерация вариантов дизайна
4	Разработка интерфейса	5 дней	Задачи 2 и 3	Помощь в создании компонентов
5	Разработка серверной части	5 дней	Задача 1	Подготовка черновиков API и кода
6	Интеграция компонентов	2 дня	Задачи 4 и 5	Выявление несоответствий интерфейсов
7	Тестирование	3 дня	Задача 6	Подготовка тестовых сценариев
8	Исправление ошибок и	2 дня	Задача 7	Классификация ошибок и

	публикация			подготовка отчёта

Обратите внимание: длительности в таблице учебные. Часть работ может выполняться параллельно, поэтому сумма длительностей отдельных задач не равна обязательно общей продолжительности проекта.

Выполните следующие действия:

- Проверьте последовательность задач и их зависимости.
- Определите, какие задачи можно выполнять параллельно.
- Укажите, какие сведения потребуются ИИ для подготовки реалистичного плана.
- Предложите, как система должна предупреждать руководителя о риске срыва срока.

Задание 3. Разработка алгоритма интеграции ИИ

Опишите, как ИИ будет взаимодействовать с системой управления проектами.

Пример схемы работы:

1. Система управления проектами

Передаёт ИИ сведения о задачах, сроках, исполнителях и статусах

2. Обработка данных

Проверка актуальности данных, анализ сроков и загрузки сотрудников

3. Анализ с помощью ИИ

Поиск рисков, прогнозирование задержек и формирование рекомендаций

4. Проверка рекомендаций

Руководитель оценивает предложенные изменения

5. Обновление проекта

После подтверждения обновляются задачи, сроки или уведомления

Ответьте на вопросы:

- Какие данные должна получать ИИ-система?
- Как часто следует проводить анализ задач?
- Что должна делать система, если обнаружен риск задержки?
- Почему нежелательно разрешать ИИ автоматически изменять все сроки без подтверждения руководителя?

Задание 4. Анализ рисков проекта

Предположим, что разработка сайта уже началась. В системе управления проектами обнаружены следующие ситуации:

Ситуация	**Возможный риск**	**Рекомендация ИИ**

Дизайн задерживается на 2 дня	Сдвиг начала разработки интерфейса	Пересмотреть последовательность работ и оценить влияние задержки
Один разработчик выполняет 6 задач одновременно	Перегрузка сотрудника	Предложить перераспределение задач
Серверная часть не прошла проверку	Задержка интеграции	Назначить проверку ошибок до начала интеграции
Требования заказчика часто меняются	Увеличение сроков и объёма работ	Фиксировать изменения и оценивать их влияние на план

Выполните задание:

- Добавьте в таблицу ещё два риска, которые могут возникнуть при разработке сайта.
- Для каждого риска определите вероятность возникновения: низкая, средняя или высокая.
- Оцените последствия риска.
- Предложите действие, которое ИИ может рекомендовать руководителю.

Для оценки используйте таблицу:

Вероятность	**Последствия**	**Приоритет**
Низкая	Незначительные	Низкий
Средняя	Умеренные	Средний
Высокая	Существенные	Высокий

Приоритет нужно обосновать: например, высокая вероятность вместе с существенными последствиями указывает на риск, который следует обработать в первую очередь.

Задание 5. Формирование отчёта с помощью ИИ

Предположим, что к концу первой недели проекта система содержит следующие данные:

Показатель	Значение
Всего задач	12
Выполнено задач	5
Выполняется	4
Не начато	3
Просрочено	2

Рассчитайте процент выполненных задач:

\frac{5}{12}\times100\%\approx41{,}7\%

Затем используйте следующий запрос в инструменте ИИ.

Проанализируй ход IT-проекта по следующим данным: всего 12 задач, выполнено 5, выполняется 4, не начато 3, просрочено 2.

- Рассчитай долю выполненных задач в процентах.
- Подготовь краткую сводку о состоянии проекта.
- Укажи проблемы, на которые руководителю следует обратить внимание.
- Предложи три действия для улучшения ситуации.
- Не делай вывод о срыве общего срока проекта, поскольку дата завершения и критический путь не указаны. Не придумывай причины просрочек.
  Скопировать запрос для ИИ

Сравните ответ ИИ с исходными данными. Проверьте расчёты и определите, какие рекомендации действительно обоснованы.

Задание 6. Оценка эффективности интеграции ИИ

Заполните таблицу, сравнив управление проектом без ИИ и с использованием ИИ.

Критерий	Без ИИ	С использованием ИИ
Составление плана	Ручное планирование	Подготовка чернового плана и зависимостей

Распределение задач	Руководитель анализирует загрузку самостоятельно	ИИ предлагает варианты распределения
Контроль сроков	Проверка статусов вручную	Автоматический поиск отклонений при настроенной интеграции
Анализ рисков	На основе опыта руководителя	Анализ данных и рекомендации ИИ
Отчётность	Ручной сбор информации	Автоматическая сводка по задачам
Ответственность за решения	Руководитель проекта	Руководитель проекта сохраняет ответственность

Сформулируйте вывод о том, какие процессы целесообразно автоматизировать в первую очередь и какие решения необходимо оставить за человеком.

4. Контрольные вопросы

Что представляет собой система управления проектами?

Какие задачи можно автоматизировать с помощью ИИ?

Какие данные необходимы ИИ для анализа хода проекта?

Как ИИ может помочь обнаружить риск срыва сроков?

Почему при составлении плана важно учитывать зависимости между задачами?

Какие ограничения существуют при использовании ИИ в управлении проектами?

Почему рекомендации ИИ должен проверять руководитель?

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Тема 1.3. Алгоритмы ИИ для обработки данных и принятия решений
Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным

контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.4. Контролировать результат обучения

Вопросы к устному опросу по теме:

1. Что такое алгоритм искусственного интеллекта? Это последовательность действий, позволяющая компьютеру анализировать данные, выявлять закономерности, делать прогнозы и помогать принимать решения.

2. Чем обучение с учителем отличается от обучения без учителя? При обучении с учителем используются данные с известными правильными ответами. При обучении без учителя алгоритм самостоятельно выявляет закономерности и группы в данных.

3. Что такое классификация данных? Это отнесение объекта к одному из заранее определённых классов. Например, определение того, является ли электронное письмо спамом.

4. Что такое кластеризация данных? Это разделение объектов на группы по сходству их характеристик без заранее заданных классов. Например, группировка покупателей по поведению.

5. Как работает алгоритм K-means? K-means разделяет данные на заданное количество кластеров, распределяя объекты по ближайшим центрам групп и постепенно уточняя эти центры.

6. Что такое регрессия и для чего она используется? Регрессия — метод моделирования зависимости числового показателя от других признаков. Она применяется, например, для прогнозирования стоимости товаров, выручки или спроса.

7. Чем классификация отличается от регрессии? Классификация предсказывает категорию объекта, например тип заявки. Регрессия предсказывает числовое значение, например стоимость товара.

8. Что такое линейная регрессия? Это метод, который описывает зависимость между признаками и целевым числовым показателем с помощью линейной формулы, например .

9. Что такое переобучение модели ИИ? Переобучение — ситуация, когда модель слишком сильно подстраивается под обучающие данные и плохо работает на новых примерах.

10. Какие методы ИИ используются для принятия решений? Применяются деревья решений, экспертные системы, модели машинного обучения, вероятностные методы и алгоритмы оптимизации.

11. Что такое система поддержки принятия решений (DSS)? DSS — информационная система, которая помогает человеку анализировать данные, сравнивать варианты и выбирать обоснованное решение.

12. Из каких основных компонентов состоит DSS? Обычно она включает подсистему данных, подсистему моделей и алгоритмов, пользовательский интерфейс, а при необходимости — базу знаний и механизм правил.

13. Как оценивается точность алгоритмов ИИ? Для классификации используют accuracy, precision и recall, а для регрессии — MAE, MSE и RMSE. Выбор метрики зависит от решаемой задачи.

14. Почему необходимо проверять результаты работы ИИ? Потому что модель может ошибаться из-за некачественных данных, переобучения или изменений условий. Проверка помогает убедиться, что результаты достоверны и подходят для практического применения.

15. Как сравнить несколько алгоритмов ИИ для решения одной задачи? Необходимо применить алгоритмы к одному набору данных, обеспечить сопоставимые условия проверки, сравнить показатели качества, скорость работы и затраты ресурсов, а затем выбрать наиболее подходящий вариант.

Критерии оценки устного опроса:

- оценка «отлично» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
- оценка «хорошо» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;

– оценка «удовлетворительно» выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
– оценка «неудовлетворительно» выставляется студенту, если он не смог изложить суть основного вопроса.

Практическая работа №10. Реализация алгоритма ИИ для анализа данных
Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.4. Контролировать результат обучения

Задание:

Задачи работы:

Изучить исходный набор данных.

Определить задачу анализа и выбрать алгоритм ИИ.

Подготовить данные для обработки.

Реализовать алгоритм на Python.

Проанализировать результаты и оценить качество модели.

Оборудование и программное обеспечение: компьютер, Python 3, среда разработки (PyCharm, Visual Studio Code или Jupyter Notebook), библиотеки pandas и scikit-learn.

2. Исходные данные

Представим, что служба технической поддержки обрабатывает обращения пользователей. Для каждой заявки известны следующие признаки:

- `priority` — приоритет заявки: 1 — низкий, 2 — средний, 3 — высокий;
- `users_affected` — количество пользователей, столкнувшихся с проблемой;
- `waiting_hours` — сколько часов заявка ожидает обработки;
- `urgent` — целевая переменная: 1 — срочная заявка, 0 — обычная.

Пример исходных данных:

№	Приоритет	Затронуто пользователей	Ожидание, часов	Срочная заявка
1	1	1	1	Нет
2	2	3	2	Нет

3	3	15	1	Да
4	2	10	6	Да
5	1	2	8	Нет
6	3	8	3	Да
7	2	4	7	Нет
8	3	20	2	Да
9	1	1	5	Нет
10	2	12	4	Да
11	3	6	1	Да
12	1	3	10	Нет

Данные учебные и специально упрощены. В реальной системе признаки и целевые классы должны формироваться на основе достоверной истории обработки заявок.

Пример решения:

Задание 1. Создание набора данных

Создайте Python-файл ai_analysis.py и добавьте исходные данные в программу.

```
import pandas as pd

data = {
    "priority": [1, 2, 3, 2, 1, 3, 2, 3, 1, 2, 3, 1],
    "users_affected": [1, 3, 15, 10, 2, 8, 4, 20, 1, 12, 6, 3],
    "waiting_hours": [1, 2, 1, 6, 8, 3, 7, 2, 5, 4, 1, 10],
    "urgent": [0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0]
}
```

```
df = pd.DataFrame(data)

print("Исходные данные:")
print(df)

print("\nПервые пять записей:")
print(df.head())

print("\nИнформация о данных:")
print(df.info())
```

Пояснение:

import pandas as pd подключает библиотеку для работы с таблицами.

data хранит признаки и целевые значения.

pd.DataFrame(data) создаёт таблицу из словаря.

df.head() выводит первые пять строк.

df.info() показывает типы данных и наличие заполненных значений.

Задание 2. Анализ и подготовка данных

Перед обучением модели необходимо отделить признаки от целевой переменной.

```
# Признаки, по которым модель делает прогноз
X = df[["priority", "users_affected", "waiting_hours"]]

# Правильные ответы для обучения
y = df["urgent"]

print("Количество заявок:", len(df))
print("Количество срочных заявок:", y.sum())
print("Количество обычных заявок:", (y == 0).sum())
```

Пояснение:

X содержит характеристики заявок, на основе которых модель будет делать прогноз.

y содержит правильные ответы, которым модель должна научиться соответствовать.

y.sum() подсчитывает количество срочных заявок, поскольку срочные заявки обозначены числом 1.

Ответьте на вопрос: почему нельзя включать столбец urgent в признаки X при обучении модели для предсказания срочности?

Задание 3. Реализация алгоритма ИИ

Разделите данные на обучающую и тестовую выборки, обучите дерево решений и получите прогнозы.

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report

# Разделение данных на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.25,
    random_state=42,
    stratify=y
)
```

```
# Создание модели
model = DecisionTreeClassifier(
    max_depth=3,
    random_state=42
)

# Обучение модели
model.fit(X_train, y_train)

# Получение прогнозов
predictions = model.predict(X_test)

print("Фактические значения:", y_test.tolist())
print("Прогнозы модели:", predictions.tolist())

# Оценка качества
accuracy = accuracy_score(y_test, predictions)

print("Accuracy:", round(accuracy, 2))
print(classification_report(
    y_test,
    predictions,
    zero_division=0
))
```

Пояснение основных команд:

train_test_split() разделяет данные на две выборки.

test_size=0.25 означает, что 25% данных выделяется для тестирования.

stratify=y помогает сохранить соотношение классов при разделении.

DecisionTreeClassifier() создаёт модель классификации.

max_depth=3 ограничивает глубину дерева, чтобы уменьшить его сложность.

model.fit() обучает модель.

model.predict() предсказывает классы для тестовых заявок.

accuracy_score() вычисляет долю правильных прогнозов.

classification_report() показывает precision, recall и F1-score по классам.

Важно: исходный набор очень маленький. Даже высокая accuracy на нём не доказывает, что модель будет хорошо работать на реальных заявках. Для надёжной оценки потребуются дополнительные данные и отдельная проверка.

Задание 4. Анализ новой заявки

Проверьте, как модель классифицирует заявку, которой не было в исходной таблице.

```
new_ticket = pd.DataFrame({
    "priority": [3],
    "users_affected": [18],
    "waiting_hours": [2]
})

prediction = model.predict(new_ticket)[0]
```

```
if prediction == 1:
    print("Модель определила заявку как срочную.")
else:
    print("Модель определила заявку как обычную.")
```

Измените значения признаков и выполните прогноз ещё для трёх заявок.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №11. Применение метода кластеризации для анализа данных

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.4. Контролировать результат обучения

Задание 1. Создание набора данных

Создайте файл clustering.py и добавьте следующий код:

```
import pandas as pd

data = {
    "customer_id": list(range(1, 13)),
    "orders": [1, 2, 1, 3, 4, 5, 6, 7, 8, 9, 2, 6],
    "average_check": [
        500, 700, 600, 900,
        1200, 1500, 1700, 2000,
        2200, 2500, 800, 1600
    ]
}

df = pd.DataFrame(data)

print("Исходные данные:")
print(df)

print("\nОбщая информация:")
print(df.info())

print("\nОсновные статистические показатели:")
print(df.describe())
```

Пояснение к коду:

- import pandas as pd подключает библиотеку для работы с таблицами.
- data хранит сведения о покупателях.

- pd.DataFrame(data) создаёт таблицу.
- df.info() показывает типы столбцов и наличие значений.
- df.describe() рассчитывает статистические показатели: среднее значение, минимум, максимум и другие характеристики.

Задание: запустите программу и определите, сколько покупателей представлено в таблице, каково минимальное и максимальное количество заказов.

Задание 2. Подготовка данных

Перед кластеризацией необходимо отделить признаки, по которым будут группироваться покупатели, от идентификатора покупателя.

```
from sklearn.preprocessing import StandardScaler

# Выбираем признаки для кластеризации
X = df[["orders", "average_check"]]

# Стандартизируем признаки
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

print("Стандартизированные данные:")
print(X_scaled)
```

Почему нужна стандартизация?

Количество заказов измеряется небольшими числами, а средний чек — сотнями и тысячами рублей. Если использовать исходные значения, денежный показатель может сильнее влиять на расстояние между объектами.

StandardScaler преобразует признаки так, чтобы каждый из них имел среднее значение около нуля и стандартное отклонение около единицы.

В результате оба признака получают сопоставимый масштаб.

Обратите внимание: customer_id не включается в кластеризацию, поскольку это идентификатор, а не характеристика поведения покупателя.

Задание 3. Реализация алгоритма K-means

Разделите покупателей на три кластера.

```
from sklearn.cluster import KMeans

model = KMeans(
    n_clusters=3,
    random_state=42,
    n_init=10
)

# Обучение модели и определение кластеров
df["cluster"] = model.fit_predict(X_scaled)

print("Результаты кластеризации:")
print(df)

print("\nКоличество покупателей в каждом кластере:")
print(df["cluster"].value_counts().sort_index())
```

Пояснение к основным параметрам:

n_clusters=3 задаёт количество групп.

random_state=42 фиксирует случайную инициализацию для воспроизводимости результата.

n_init=10 запускает алгоритм с несколькими начальными вариантами и выбирает лучший результат по внутреннему критерию K-means.

fit_predict() выполняет кластеризацию и возвращает номер кластера для каждого покупателя.

value_counts() подсчитывает количество покупателей в каждой группе.

Номера кластеров 0, 1 и 2 условны. Нельзя считать, что кластер 2 обязательно лучше кластера 0: номера не отражают рейтинг или ценность группы.

Задание 4. Анализ полученных кластеров

После группировки необходимо определить характеристики каждой группы.

```
cluster_stats = df.groupby("cluster").agg(
   customers=("customer_id", "count"),
   avg_orders=("orders", "mean"),
   avg_check=("average_check", "mean")
).round(2)

print("Характеристики кластеров:")
print(cluster_stats)
```

Код рассчитывает для каждой группы:

количество покупателей;

среднее количество заказов;

средний чек.

Используя полученную таблицу, заполните аналитическую таблицу.

Кластер	**Среднее количество заказов**	**Средний чек**	**Интерпретация**
0	По результатам программы	По результатам программы	Определить после анализа
1	По результатам программы	По результатам программы	Определить после анализа
2	По результатам программы	По результатам программы	Определить после анализа

Например, если в одной группе среднее количество заказов и средний чек выше, чем в остальных, её можно предварительно интерпретировать как группу активных покупателей с высокой средней суммой покупки.

Однако нельзя автоматически считать такую группу наиболее прибыльной: для оценки прибыли потребуются данные о себестоимости, скидках, возвратах и других расходах.

Задание 5. Визуализация кластеров

Для наглядного представления результатов постройте диаграмму рассеяния.

```
import matplotlib.pyplot as plt

plt.figure(figsize=(9, 6))

for cluster_id in sorted(df["cluster"].unique()):
    group = df[df["cluster"] == cluster_id]

    plt.scatter(
        group["orders"],
        group["average_check"],
        label=f"Кластер {cluster_id}"
    )

plt.xlabel("Количество заказов")
plt.ylabel("Средний чек, руб.")
plt.title("Кластеризация покупателей методом K-means")
plt.legend()
plt.grid(True)
plt.show()
```

На диаграмме:

горизонтальная ось показывает количество заказов;

вертикальная ось показывает средний чек;

цвет и легенда позволяют различать кластеры.

Задание: проанализируйте диаграмму и ответьте, насколько хорошо разделились группы. Есть ли покупатели, характеристики которых близки к характеристикам сразу нескольких кластеров?

Задание 6. Оценка качества кластеризации

Для предварительной оценки используйте силуэтный коэффициент. Он показывает, насколько объект похож на собственный кластер по сравнению с другими кластерами.

```
from sklearn.metrics import silhouette_score

score = silhouette_score(X_scaled, df["cluster"])

print("Силуэтный коэффициент:", round(score, 3))
```

Общая интерпретация:

Значение коэффициента	Возможная интерпретация

Близко к 1	Кластеры хорошо разделены
Около 0	Кластеры пересекаются или находятся близко друг к другу
Меньше 0	Некоторые объекты могут быть отнесены к неподходящим кластерам

Эти ориентиры не являются абсолютными правилами. Качество кластеризации нужно оценивать вместе с характеристиками данных и практическим смыслом полученных групп.

Задание 7. Сравнение количества кластеров

Проверьте, как меняется результат при использовании двух, трёх и четырёх кластеров.

```
for k in [2, 3, 4]:
    model = KMeans(
        n_clusters=k,
        random_state=42,
        n_init=10
    )

    labels = model.fit_predict(X_scaled)

    score = silhouette_score(X_scaled, labels)

    print(
        f"Количество кластеров: {k}, "
        f"силуэтный коэффициент: {score:.3f}"
    )
```

Сравните полученные значения и ответьте:

При каком количестве кластеров силуэтный коэффициент оказался наибольшим?

Совпадает ли лучший результат по коэффициенту с наиболее понятным для бизнеса разделением?

Почему нельзя выбирать количество кластеров только по одному показателю?

Помните, что максимальный силуэтный коэффициент не всегда означает, что найденное разделение наиболее полезно для практической задачи.

4. Дополнительное задание повышенной сложности

Измените исходный набор данных и добавьте два признака:

returns — количество возвратов за месяц;

days_since_purchase — количество дней с момента последней покупки.

Повторите кластеризацию с использованием четырёх признаков:

orders, average_check, returns, days_since_purchase.

В отчёте укажите:

Изменились ли размеры и состав кластеров?

Какие новые закономерности удалось обнаружить?

Как дополнительная информация о возвратах и давности покупки влияет на сегментацию?

Какие решения интернет-магазин может принимать на основе полученных групп?

При подготовке признаков необходимо учитывать их масштаб, а при интерпретации результатов — проверять, действительно ли выделенные группы полезны для бизнеса.

5. Контрольные вопросы

Что такое кластеризация и для чего она используется?

Чем кластеризация отличается от классификации?

Как работает алгоритм K-means?

Что обозначает параметр n_clusters?

Почему перед кластеризацией может потребоваться стандартизация данных?

Для чего нужен идентификатор объекта и почему его обычно не включают в признаки?

Что показывает силуэтный коэффициент?

Почему результат кластеризации может измениться при выборе другого количества кластеров?

Какие преимущества и ограничения есть у K-means?

Как можно применить результаты кластеризации покупателей в интернет-магазине?

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №12. Применение регрессионных методов для предсказаний
Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.4. Контролировать результат обучения

Задание:

Разработать программу на языке Python для прогнозирования числовых значений с использованием регрессионных методов машинного обучения.

- Создать набор данных, содержащий входные признаки и целевой числовой показатель для прогнозирования.
- Подготовить данные и разделить их на обучающую и тестовую выборки.
- Реализовать и обучить модель линейной регрессии с использованием библиотеки scikit-learn.
- Получить предсказания модели на тестовой выборке.
- Оценить качество прогнозирования с помощью метрик MAE, RMSE и R^2.
- Построить график сравнения фактических и прогнозируемых значений.
- Использовать обученную модель для прогнозирования нового значения.
- Реализовать вторую модель регрессии — дерево регрессии — и сравнить результаты двух алгоритмов.
- Проанализировать полученные результаты и сделать вывод о качестве моделей.

Пример решения:

Рассмотрим задачу прогнозирования объёма продаж компании в зависимости от рекламного бюджета. Для решения задачи используем линейную регрессию и дерево регрессии.

1. Подключение библиотек и создание данных

```
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score

data = {
   "advertising": [
      10, 15, 20, 25, 30,
      35, 40, 45, 50, 55,
      60, 65, 70, 75, 80,
      85, 90, 95, 100, 105
   ],
   "sales": [
      22, 28, 31, 37, 39,
      45, 48, 54, 57, 61,
      65, 71, 73, 80, 83,
      87, 94, 96, 103, 106
   ]
}

df = pd.DataFrame(data)

print(df)
```

В таблице advertising обозначает рекламный бюджет в тысячах рублей, а sales — объём продаж в тысячах рублей.

2. Подготовка данных

```
X = df[["advertising"]]
y = df["sales"]
```

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42
)
```

Переменная X содержит входной признак, а y — целевой показатель. Для обучения модели используется 80% данных, остальные 20% предназначены для проверки качества прогнозирования.

3. Обучение модели линейной регрессии

```
model = LinearRegression()

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

print("Фактические значения:", y_test.to_list())
print("Прогнозы:", y_pred.round(2))
```

Метод fit() обучает модель на известных данных, а predict() рассчитывает прогнозируемые значения для тестовой выборки.

4. Оценка качества модели

```
mae = mean_absolute_error(y_test, y_pred)

rmse = mean_squared_error(
    y_test, y_pred
) ** 0.5

r2 = r2_score(y_test, y_pred)

print("MAE:", round(mae, 2))
print("RMSE:", round(rmse, 2))
print("R²:", round(r2, 3))
```

Метрика MAE показывает среднюю абсолютную ошибку, RMSE сильнее учитывает большие ошибки, а R^2 характеризует соответствие модели данным. Результаты необходимо оценивать в совокупности.

5. Построение графика

```
plt.scatter(
    X_test["advertising"],
    y_test,
    label="Фактические значения"
)

plt.scatter(
    X_test["advertising"],
    y_pred,
    label="Прогноз модели"
)

plt.xlabel("Рекламный бюджет, тыс. руб.")
plt.ylabel("Продажи, тыс. руб.")
plt.title("Прогнозирование объёма продаж")
```

```
plt.legend()
plt.grid(True)
plt.show()
```

На графике отображаются фактические продажи и результаты прогнозирования. Чем ближе прогнозируемые значения к фактическим, тем меньше ошибки модели на тестовых данных.

6. Прогнозирование нового значения

```
new_data = pd.DataFrame({
    "advertising": [50]
})

prediction = model.predict(new_data)

print(
    "Прогноз продаж:",
    round(prediction[0], 2),
    "тыс. рублей"
)
```

Программа прогнозирует объём продаж при рекламном бюджете 50 тысяч рублей.

7. Реализация дерева регрессии

```
tree_model = DecisionTreeRegressor(
    max_depth=3,
    random_state=42
)

tree_model.fit(X_train, y_train)

tree_pred = tree_model.predict(X_test)

print(
    "MAE дерева:",
    round(mean_absolute_error(y_test, tree_pred), 2)
)

print(
    "RMSE дерева:",
    round(mean_squared_error(y_test, tree_pred) ** 0.5, 2)
)

print(
    "R² дерева:",
    round(r2_score(y_test, tree_pred), 3)
)
```

После выполнения программы необходимо сравнить метрики обеих моделей. Модель с меньшими значениями MAE и RMSE и, как правило, большим значением R^2 лучше справляется с прогнозированием на выбранной тестовой выборке.

Вывод: в результате выполнения задания реализованы две регрессионные модели, выполнено прогнозирование числовых значений и проведена оценка качества алгоритмов. Полученные результаты позволяют сравнить модели и определить, какая из них лучше подходит для рассматриваемой задачи на имеющихся данных.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №13. Внедрение модели ИИ в систему поддержки принятия решений

Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.4. Контролировать результат обучения

Задание:

Разработать программу, реализующую простейшую систему поддержки принятия решений (СППР) с использованием модели искусственного интеллекта на языке Python.

Система должна анализировать входные данные, применять обученную модель машинного обучения и формировать рекомендации для пользователя на основе полученного результата.

Для выполнения работы необходимо:

Выбрать предметную область, в которой требуется принимать решения. Например, оценка риска просрочки заказа, определение приоритета обращения в службу поддержки или оценка вероятности отказа оборудования.

Подготовить набор данных для обучения модели ИИ.

Реализовать модель классификации с использованием библиотеки scikit-learn.

Обучить модель на подготовленных данных.

Разработать программный модуль, который принимает новые данные и передаёт их обученной модели.

Реализовать механизм формирования рекомендаций на основе результата классификации.

Организовать вывод исходных данных, результата работы модели и рекомендуемого действия.

Проверить работу системы на нескольких примерах и проанализировать полученные результаты.

Пример решения:

Рассмотрим задачу создания системы поддержки принятия решений для службы технической поддержки. Система должна анализировать поступающие обращения и определять, требуется ли их срочная обработка.

Для каждого обращения учитываются следующие признаки:

priority — приоритет обращения: 1 — обычный, 2 — высокий, 3 — критический;

users_affected — количество пользователей, столкнувшихся с проблемой;

waiting_hours — время ожидания решения в часах;

urgent — целевой класс: 1 — обращение срочное, 0 — обращение обычное.

На основе этих данных модель классификации будет определять категорию нового обращения, а система — предлагать соответствующее действие.

1. Подключение библиотек и создание набора данных

```
import pandas as pd

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

data = {
    "priority": [
        1, 1, 2, 3, 2, 3, 1, 2, 3, 1,
        2, 3, 1, 2, 3, 1, 2, 3, 2, 1
    ],
    "users_affected": [
        1, 2, 10, 50, 15, 100, 3, 20, 80, 1,
        25, 120, 4, 18, 60, 2, 30, 90, 12, 5
    ],
    "waiting_hours": [
        1, 2, 3, 1, 5, 2, 4, 6, 1, 2,
        8, 3, 1, 7, 2, 5, 4, 1, 6, 3
    ],
    "urgent": [
        0, 0, 0, 1, 1, 1, 0, 1, 1, 0,
        1, 1, 0, 1, 1, 0, 1, 1, 1, 0
    ]
}

df = pd.DataFrame(data)

print(df)
```

Создаётся таблица с примерами обращений. В реальной системе данные могут поступать из базы данных или информационной системы технической поддержки. В рамках практической работы используется учебный набор данных.

2. Подготовка данных и обучение модели

```
X = df[
    ["priority", "users_affected", "waiting_hours"]
]

y = df["urgent"]

X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.25,
    random_state=42,
    stratify=y
)
```

```
model = DecisionTreeClassifier(
    max_depth=3,
    random_state=42
)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print("Точность модели:", round(accuracy, 2))
```

Переменная X содержит признаки обращений, а y — известные классы, которые модель должна научиться определять.

Метод fit() выполняет обучение, а predict() позволяет классифицировать обращения, которые не использовались для обучения. Функция accuracy_score() рассчитывает долю правильных предсказаний на тестовой выборке.

3. Создание механизма поддержки принятия решений

После обучения модели необходимо разработать функцию, которая не только определяет класс обращения, но и предлагает сотруднику рекомендуемое действие.

```
def make_decision(priority, users_affected, waiting_hours):
    new_ticket = pd.DataFrame([{
        "priority": priority,
        "users_affected": users_affected,
        "waiting_hours": waiting_hours
    }])

    prediction = model.predict(new_ticket)[0]

    print("\nРезультат анализа обращения")
    print("Приоритет:", priority)
    print("Количество затронутых пользователей:", users_affected)
    print("Время ожидания:", waiting_hours, "ч.")

    if prediction == 1:
        recommendation = (
            "Передать обращение специалисту "
            "для срочной обработки"
        )
        category = "Срочное обращение"
    else:
        recommendation = (
            "Обработать обращение в обычном порядке"
        )
        category = "Обычное обращение"

    print("Категория:", category)
    print("Рекомендация:", recommendation)

    return prediction
```

Функция make_decision() принимает характеристики обращения, формирует таблицу нужного формата и передаёт её модели. После получения предсказания система выбирает рекомендацию.

Таким образом, модель ИИ отвечает за классификацию, а программная логика СППР преобразует результат классификации в понятное пользователю действие.

4. Проверка работы системы

Вызовем функцию для нескольких обращений с разными характеристиками.

```
make_decision(
   priority=3,
   users_affected=80,
   waiting_hours=2
)

make_decision(
   priority=1,
   users_affected=2,
   waiting_hours=1
)

make_decision(
   priority=2,
   users_affected=20,
   waiting_hours=7
)
```

Для каждого обращения программа выведет входные параметры, категорию, определённую моделью, и рекомендуемое действие.

Результаты могут различаться в зависимости от того, какие закономерности модель выявила во время обучения. Поскольку учебный набор небольшой и искусственно составлен, полученные рекомендации нельзя считать готовым решением для реальной службы поддержки.

5. Дополнительное развитие системы

Для расширения функциональности СППР можно:

добавить сохранение результатов анализа в CSV-файл;

реализовать ввод параметров обращения с клавиатуры;

добавить новые признаки, например тип проблемы или критичность информационной системы;

вывести вероятность принадлежности обращения к каждому классу с помощью predict_proba();

реализовать отдельные рекомендации для разных категорий обращений;

сравнить дерево решений с алгоритмом случайного леса (RandomForestClassifier).

Вывод: в ходе выполнения практической работы создан прототип системы поддержки принятия решений с использованием модели искусственного интеллекта. Модель классифицирует обращения, а программный модуль формирует рекомендации для пользователя. Реализация демонстрирует, как алгоритмы машинного обучения могут интегрироваться в информационные системы для автоматизации анализа данных и поддержки принятия решений.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.

– Оформление отчёта — 1 балл.

Практическая работа №14. Сравнение различных алгоритмов ИИ на одном наборе данных
Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.3. Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта
ПК 3.4. Контролировать результат обучения

Задание:

Разработать программу на языке Python для сравнительного анализа нескольких алгоритмов машинного обучения на одном наборе данных.

Для выполнения работы необходимо:

Выбрать задачу классификации и подготовить набор данных с входными признаками и целевым классом.

Разделить набор данных на обучающую и тестовую выборки.

Реализовать не менее трёх алгоритмов машинного обучения, например:

- логистическую регрессию;
- дерево решений;
- случайный лес.

Обучить каждый алгоритм на одной и той же обучающей выборке.

Получить предсказания всех моделей на одной и той же тестовой выборке.

Рассчитать показатели качества классификации: accuracy, precision, recall и F1-score.

Сохранить результаты сравнения в таблицу.

Построить график, позволяющий сравнить точность алгоритмов.

Проанализировать результаты и определить, какой алгоритм показал лучшие результаты на выбранном наборе данных.

Пример решения:

Рассмотрим задачу классификации обращений в службу технической поддержки. Необходимо определить, является ли обращение срочным, на основе его приоритета, количества затронутых пользователей и времени ожидания решения.

Для решения задачи сравним три алгоритма машинного обучения: логистическую регрессию, дерево решений и случайный лес.

1. Подключение библиотек и создание набора данных

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import (
    accuracy_score,
    precision_score,
    recall_score,
    f1_score
)

data = {
    "priority": [
        1, 1, 2, 3, 2, 3, 1, 2, 3, 1,
        2, 3, 1, 2, 3, 1, 2, 3, 2, 1,
        3, 2, 1, 3, 2, 1, 3, 2, 1, 3
    ],
    "users_affected": [
        1, 2, 10, 50, 15, 100, 3, 20, 80, 1,
        25, 120, 4, 18, 60, 2, 30, 90, 12, 5,
        70, 22, 6, 110, 16, 3, 85, 28, 4, 95
    ],
    "waiting_hours": [
        1, 2, 3, 1, 5, 2, 4, 6, 1, 2,
        8, 3, 1, 7, 2, 5, 4, 1, 6, 3,
        2, 5, 2, 4, 7, 1, 3, 6, 2, 1
    ],
    "urgent": [
        0, 0, 0, 1, 1, 1, 0, 1, 1, 0,
        1, 1, 0, 1, 1, 0, 1, 1, 1, 0,
        1, 1, 0, 1, 1, 0, 1, 1, 0, 1
    ]
}

df = pd.DataFrame(data)

print(df)
```

В наборе данных представлены 30 обращений. Признаки описывают характеристики обращений, а столбец urgent содержит целевой класс: 1 — срочное обращение, 0 — обычное.

2. Подготовка данных

```
X = df[
    ["priority", "users_affected", "waiting_hours"]
]

y = df["urgent"]

X_train, X_test, y_train, y_test = train_test_split(
    X,
    y,
    test_size=0.3,
    random_state=42,
```

```
    stratify=y
)
```

Переменная X содержит входные признаки, а y — целевой класс.

Метод train_test_split() разделяет данные на обучающую и тестовую выборки. В данном случае 70% записей используются для обучения, а 30% — для оценки качества моделей.

Параметр stratify=y помогает сохранить соотношение классов при разделении данных. Параметр random_state=42 обеспечивает воспроизводимость эксперимента.

3. Создание моделей машинного обучения

```
models = {
    "Логистическая регрессия": make_pipeline(
        StandardScaler(),
        LogisticRegression(max_iter=1000)
    ),

    "Дерево решений": DecisionTreeClassifier(
        max_depth=4,
        random_state=42
    ),

    "Случайный лес": RandomForestClassifier(
        n_estimators=100,
        max_depth=4,
        random_state=42
    )
}
```

В программе используются три алгоритма:

Логистическая регрессия — классифицирует данные, оценивая вероятность принадлежности объекта к классу.

Дерево решений — последовательно применяет условия к признакам и на их основе определяет класс.

Случайный лес — объединяет результаты нескольких деревьев решений, чтобы получить итоговое предсказание.

Для логистической регрессии используется StandardScaler(), который приводит признаки к сопоставимому масштабу. Деревья решений и случайный лес обычно не требуют такой стандартизации.

4. Обучение моделей и оценка результатов

```
results = []

for name, model in models.items():
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)

    results.append({
        "Алгоритм": name,
        "Accuracy": accuracy_score(y_test, y_pred),
        "Precision": precision_score(
            y_test, y_pred, zero_division=0
        ),
        "Recall": recall_score(
```

```
            y_test, y_pred, zero_division=0
        ),
        "F1-score": f1_score(
            y_test, y_pred, zero_division=0
        )
    })

results_df = pd.DataFrame(results)

print(results_df.round(3))
```

Каждая модель обучается на одной и той же обучающей выборке, после чего проверяется на одинаковых тестовых данных.

Используются четыре метрики:

Accuracy — доля правильно классифицированных объектов.

Precision — доля действительно срочных обращений среди всех обращений, которые модель назвала срочными.

Recall — доля найденных срочных обращений среди всех действительно срочных.

F1-score — гармоническое среднее precision и recall.

Для службы поддержки особенно важно учитывать Recall: низкое значение может означать, что модель пропускает много срочных обращений.

5. Построение графика сравнения алгоритмов

```
plt.figure(figsize=(9, 5))

plt.bar(
    results_df["Алгоритм"],
    results_df["Accuracy"]
)

plt.title("Сравнение точности алгоритмов ИИ")
plt.xlabel("Алгоритм")
plt.ylabel("Accuracy")

plt.ylim(0, 1)
plt.xticks(rotation=15)
plt.grid(axis="y")

plt.tight_layout()
plt.show()
```

На графике отображается точность каждого алгоритма на тестовой выборке. Чем выше столбец, тем большую долю объектов модель классифицировала правильно.

При этом высокая accuracy сама по себе не гарантирует, что алгоритм хорошо выявляет срочные обращения. Поэтому необходимо также учитывать precision, recall и F1-score.

6. Определение лучшей модели

```
best_model = results_df.sort_values(
    by="F1-score",
    ascending=False
).iloc[0]
```

```
print("Лучший алгоритм по F1-score:")
print(best_model["Алгоритм"])

print("F1-score:", round(best_model["F1-score"], 3))
print("Accuracy:", round(best_model["Accuracy"], 3))
```

Программа выбирает алгоритм с наибольшим значением F1-score среди проверенных моделей. Такой критерий позволяет учитывать одновременно precision и recall.

Полученный результат относится только к выбранному набору данных и конкретному разделению на обучающую и тестовую выборки. На небольшом учебном наборе показатели могут существенно меняться, поэтому нельзя делать вывод, что выбранный алгоритм всегда будет лучше остальных.

7. Дополнительное задание

Измените программу и выполните следующие действия:

Добавьте четвёртый алгоритм — метод ближайших соседей KNeighborsClassifier.

Сравните результаты при разных значениях параметра n_neighbors.

Постройте график сравнения не только accuracy, но и F1-score.

Измените размер тестовой выборки на 20% и повторите эксперимент.

Объясните, почему результаты моделей изменяются при изменении состава данных.

Определите, какой алгоритм предпочтительнее для задачи выявления срочных обращений, учитывая необходимость не пропускать критические случаи.

Вывод: в ходе выполнения практической работы реализованы и сравнены три алгоритма машинного обучения на одном наборе данных. Рассчитаны основные метрики качества классификации, построен график и определена модель с наилучшим результатом по F1-score. Работа позволяет изучить влияние выбора алгоритма на качество решения прикладной задачи и понять, почему при оценке моделей необходимо использовать несколько метрик.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Тема 1.4. Этические и правовые аспекты использования ИИ
Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.4. Контролировать результат обучения
ПК 3.5. Оформлять результат проведения процедуры обучения

Вопросы к устному опросу по теме:

1. Что понимается под этическими аспектами использования искусственного интеллекта?

Этические аспекты использования ИИ — это принципы и нормы, определяющие, насколько ответственным, справедливым и безопасным является применение интеллектуальных систем. К основным этическим принципам относятся: Справедливость — отсутствие необоснованной дискриминации пользователей. Прозрачность — возможность понять, на каких основаниях система принимает решения. Ответственность — наличие лиц, отвечающих за применение и последствия работы ИИ.

Безопасность — предотвращение вреда пользователям и организациям.

1. Защита частной жизни — соблюдение конфиденциальности персональных данных.

Например, если ИИ используется для отбора кандидатов на работу, он не должен необоснованно отдавать предпочтение определённым группам людей.

2. Какие этические проблемы могут возникать при использовании ИИ в информационных системах?

Основные проблемы связаны с предвзятостью алгоритмов, нарушением конфиденциальности, недостаточной прозрачностью решений и чрезмерной автоматизацией.

Например, модель, обученная на исторических данных компании, может воспроизводить существовавшие ранее ошибки и предубеждения. Генеративный ИИ также способен создавать недостоверную информацию, которая выглядит убедительно.

Для уменьшения этих рисков необходимо проверять данные, тестировать алгоритмы на разных группах пользователей, контролировать результаты и предусматривать возможность вмешательства человека.

3. Что такое алгоритмическая предвзятость и почему она опасна?

Алгоритмическая предвзятость — это систематическое смещение результатов работы ИИ, из-за которого определённые пользователи или группы получают менее благоприятные результаты.

Причинами могут быть:

1. неполные или несбалансированные обучающие данные;
2. ошибки при выборе признаков;
3. некорректная постановка задачи;
4. исторические предубеждения, отражённые в данных.

Например, система оценки заёмщиков может чаще отклонять заявки определённой группы клиентов из-за особенностей данных, на которых она обучалась.

Предвзятость опасна тем, что автоматизированное решение может затрагивать большое количество людей. Для её выявления применяют сравнительное тестирование, анализ ошибок и аудит модели.

4. Какие правовые аспекты необходимо учитывать при внедрении ИИ в информационные системы?

При внедрении ИИ необходимо учитывать законодательство о персональных данных, интеллектуальной

собственности, защите информации, договорных отношениях и отраслевые требования.

В России особое значение имеют:

1. Федеральный закон № 152-ФЗ «О персональных данных»;
2. Федеральный закон № 149-ФЗ «Об информации, информационных технологиях и о защите информации»;
3. нормы Гражданского кодекса РФ об интеллектуальной собственности;
4. законодательство об информации ограниченного доступа и коммерческой тайне, когда оно применимо.

Кроме того, организация должна учитывать условия лицензий на программное обеспечение, договоры с поставщиками ИИ-сервисов и правила обработки информации.

Применимые требования зависят от назначения системы, характера данных и конкретного сценария использования ИИ.

5. Кто несёт ответственность за ошибки искусственного интеллекта?

Ответственность определяется обстоятельствами происшествия, применимым законодательством, договорными условиями и ролью участников.

В зависимости от ситуации значение могут иметь действия разработчика, организации — оператора системы, поставщика программного обеспечения или сотрудника, использующего результат ИИ.

Например, если организация применяет ИИ для автоматической проверки документов и из-за ошибки незаконно отказывает клиенту в услуге, сам факт автоматизации не освобождает организацию от возможной ответственности.

Поэтому необходимо назначать ответственных лиц, документировать решения, проверять качество системы и обеспечивать возможность обжалования значимых результатов.

6. Какие требования предъявляются к защите персональных данных при использовании ИИ?

При обработке персональных данных необходимо соблюдать требования законодательства, прежде всего Федерального закона № 152-ФЗ.

К основным мерам относятся:

1. наличие законного основания обработки данных;
2. определение конкретных целей обработки;
3. сбор только необходимых данных;
4. обеспечение конфиденциальности и безопасности;
5. ограничение доступа сотрудников;
6. установление сроков хранения и правил удаления данных;
7. выполнение требований к локализации и передаче данных, если они применимы.

Например, перед передачей клиентских обращений во внешний ИИ-сервис организация должна проверить, допустима ли такая передача, какие данные передаются, где они обрабатываются и как

поставщик обеспечивает их защиту.

7. Какие риски возникают при использовании ИИ в информационных системах?

Риски использования ИИ можно разделить на несколько групп:

Вид риска	Пример
Технический	Модель выдаёт неправильные результаты
Информационный	ИИ раскрывает конфиденциальные сведения
Правовой	Обработка данных осуществляется без надлежащего основания
Этический	Алгоритм необоснованно дискриминирует пользователей
Кибербезопасности	Злоумышленник манипулирует входными данными
Организационный	Сотрудники без проверки полагаются на выводы модели
Финансовый	Ошибочные прогнозы приводят к убыткам

Для управления рисками необходимо проводить тестирование, устанавливать ограничения на использование системы, контролировать доступ к данным и регулярно проверять качество работы модели.

8. Какие последствия могут возникнуть из-за ошибок в работе ИИ?

Последствия зависят от области применения системы. В бизнесе ошибки могут привести к финансовым потерям, в образовании — к необоснованной оценке студентов, в медицине — к неправильной интерпретации информации, а в информационной безопасности — к пропуску угроз.

Ошибки также могут привести к нарушению прав граждан, утечкам данных, репутационным потерям и юридическим спорам.

Для снижения последствий применяют тестирование на реальных сценариях, мониторинг ошибок, резервные процедуры, журналирование событий и обязательную проверку человеком решений с

существенными последствиями.

9. Какие нормативно-правовые документы регулируют использование ИИ в России?

В России регулирование ИИ складывается из общих законодательных норм, специальных отраслевых требований и документов стратегического характера.

К важным документам относятся:

1. Федеральный закон № 152-ФЗ «О персональных данных»;
2. Федеральный закон № 149-ФЗ «Об информации, информационных технологиях и о защите информации»;
3. Гражданский кодекс РФ — в части интеллектуальной собственности и договорных отношений;
4. Указ Президента РФ от 10 октября 2019 г. № 490 «О развитии искусственного интеллекта в Российской Федерации», которым утверждена Национальная стратегия развития искусственного интеллекта;
5. иные нормативные акты, применимые к конкретной отрасли и сценарию использования технологии.

Важно различать обязательные нормы законодательства и стратегии, рекомендации и стандарты: они имеют разный юридический статус. Перед внедрением ИИ необходимо проверять актуальную редакцию документов и учитывать специальные требования к конкретной деятельности.

10. Как регулируется использование искусственного интеллекта в других странах?

Подходы к регулированию ИИ различаются.

В Европейском союзе действует Регламент ЕС об искусственном интеллекте — AI Act. Он предусматривает требования, зависящие от уровня риска системы, а для отдельных видов ИИ устанавливает специальные ограничения и обязанности.

В США регулирование складывается из федеральных и региональных норм, отраслевых требований и рекомендаций государственных органов. В разных штатах могут действовать собственные правила в отношении отдельных сценариев применения ИИ.

Международные организации также разрабатывают принципы ответственного использования ИИ. Например, рекомендации ОЭСР и документы ЮНЕСКО затрагивают вопросы прозрачности, безопасности, справедливости и защиты прав человека.

При работе с зарубежными клиентами или данными организация должна учитывать законодательство соответствующих юрисдикций, а не только нормы своей страны.

11. Что такое прозрачность искусственного интеллекта и объяснимость его решений?

Прозрачность означает, что пользователю и ответственным специалистам доступна необходимая информация о назначении, ограничениях и условиях применения системы.

Объяснимость — возможность понять, какие факторы повлияли на конкретный результат модели и почему было принято определённое решение.

Например, кредитная организация может использовать модель для оценки риска. Объяснимость позволяет выяснить, какие параметры заявки существенно повлияли на оценку клиента.

Для обеспечения прозрачности применяют документацию модели, описание обучающих данных, журналирование решений, визуализацию значимых признаков и методы объяснения предсказаний. При

этом объяснение модели не всегда является полным доказательством причинно-следственной связи.

12. Как можно использовать ИИ для мониторинга соблюдения правовых норм?

ИИ может анализировать документы, договоры, отчёты и операции, выявляя признаки возможных нарушений.

Примеры применения:

1. поиск отсутствующих обязательных пунктов в договорах;
2. обнаружение подозрительных финансовых операций;
3. проверка документов на соответствие установленным требованиям;
4. выявление аномалий в журналах информационных систем;
5. анализ соблюдения внутренних политик безопасности.

Например, система может проверять договоры на наличие обязательных реквизитов и формировать список документов, требующих дополнительной проверки.

Однако результат ИИ следует рассматривать как инструмент предварительного анализа. Обнаруженный системой риск не всегда означает, что нарушение действительно произошло. Значимые выводы необходимо проверять с учётом контекста и требований законодательства.

13. Как организовать защиту данных в информационной системе с использованием ИИ?

Система защиты данных с использованием ИИ может включать несколько взаимосвязанных компонентов:

1. Сбор журналов событий и информации о действиях пользователей.
2. Предварительную обработку и фильтрацию данных.
3. Анализ поведения пользователей и сетевой активности.
4. Выявление аномалий с помощью алгоритмов машинного обучения.
5. Формирование предупреждений для специалистов по информационной безопасности.
6. Регистрацию результатов анализа и действий по реагированию.

Например, модель может обнаружить необычно большое количество запросов к базе данных со стороны одной учётной записи и передать событие специалисту для проверки.

При этом необходимо защищать и саму систему ИИ: ограничивать доступ к модели, контролировать источники данных, предотвращать утечки и регулярно проверять корректность обнаружения угроз.

14. Какие методы применяются для оценки и снижения рисков использования ИИ?

Оценка рисков начинается с определения возможного вреда, вероятности его возникновения и последствий для пользователей и организации.

Основные методы:

1. Анализ сценариев — изучение ситуаций, в которых система может ошибиться.
2. Тестирование модели — проверка точности и устойчивости на разных наборах данных.
3. Аудит данных — выявление пропусков, ошибок и возможной предвзятости.
4. Оценка воздействия на защиту данных — анализ рисков обработки персональной информации в случаях, когда такая оценка требуется или необходима для управления рисками.
5. Контроль человеком — проверка значимых решений специалистом.
6. Мониторинг после внедрения — отслеживание ошибок и изменения качества модели.

Результаты оценки оформляются в виде перечня рисков, их последствий, вероятности возникновения, мер

предотвращения и ответственных лиц.

15. Какие этапы необходимо выполнить при подготовке отчёта о соблюдении законодательства при внедрении ИИ?

Отчёт должен показывать, какие требования применимы к системе, как организация их выполняет и какие риски остаются.

Основные этапы подготовки:

1. Описание системы — назначение ИИ, область применения, категории пользователей и используемые данные.
2. Определение требований — выявление применимых законов, нормативных актов, договорных условий и внутренних политик.
3. Анализ обработки данных — проверка целей, оснований, объёма и сроков хранения информации.
4. Оценка рисков — определение возможных нарушений, ошибок модели и последствий для пользователей.
5. Проверка мер защиты — анализ доступа, журналирования, резервного копирования и других организационных и технических мер.
6. Формирование рекомендаций — описание необходимых исправлений, сроков и ответственных лиц.
7. Подготовка заключения — фиксация проверенных требований, выявленных несоответствий и дальнейших действий.

Отчёт должен опираться на подтверждающие материалы: документацию системы, результаты проверок, внутренние регламенты и актуальные нормативные документы. Наличие такого отчёта само по себе не гарантирует полного соблюдения законодательства, но помогает систематизировать проверку и выявить проблемные области.

Критерии оценки устного опроса:

- оценка «отлично» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов и смог ответить на уточняющие вопросы;
- оценка «хорошо» выставляется студенту, если он смог изложить суть основного вопроса, смог ответить по существу дополнительных вопросов, но не смог ответить на уточняющие вопросы;
- оценка «удовлетворительно» выставляется студенту, если он смог изложить суть основного вопроса, но не смог ответить по существу дополнительных вопросов;
- оценка «неудовлетворительно» выставляется студенту, если он не смог изложить суть основного вопроса.

Практическая работа №15. Анализ кейсов этических вопросов в ИИ
Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.4. Контролировать результат обучения
ПК 3.5. Оформлять результат проведения процедуры обучения

Задание:

Провести анализ практических ситуаций, связанных с этическими проблемами применения искусственного интеллекта в информационных системах, выявить возможные нарушения этических

принципов и предложить способы их предотвращения.

Пример решения: Рассмотрим три ситуации применения ИИ: автоматический отбор кандидатов на работу, использование чат-бота для обработки обращений клиентов и применение ИИ для распознавания лиц. 1. Кейс № 1. Предвзятость при автоматическом отборе кандидатов. Описание ситуации.

Компания использует систему искусственного интеллекта для автоматического анализа резюме соискателей. Модель обучалась на данных о кандидатах, которых компания принимала на работу в предыдущие годы. Через некоторое время выяснилось, что система чаще отклоняет резюме определённой группы кандидатов, даже если их профессиональные навыки соответствуют требованиям вакансии.

Анализ проблемы.

Возможная причина заключается в том, что обучающие данные отражают исторические особенности найма и могли содержать скрытые закономерности, связанные с необоснованными предпочтениями. Модель могла усвоить эти закономерности и воспроизводить их при обработке новых резюме.

Нарушается принцип справедливости, поскольку кандидаты могут получать разные результаты не из-за профессиональных качеств, а вследствие особенностей данных и алгоритма.

Возможные последствия:

1. необоснованный отказ подходящим кандидатам;
2. снижение разнообразия сотрудников;
3. репутационные потери компании;
4. риск нарушения применимых требований законодательства.

Предлагаемые решения:

1. Проверить обучающие данные на наличие предвзятости.
2. Оценить качество модели на разных группах кандидатов.
3. Исключить признаки, не связанные с профессиональными требованиями, если их использование не имеет законного и обоснованного основания.
4. Предусмотреть проверку результатов специалистом по подбору персонала.
5. Регулярно проводить аудит модели после внедрения.

Вывод по кейсу: автоматизация найма не должна приводить к необоснованной дискриминации. Решения модели необходимо контролировать и проверять на справедливость.

2. Кейс № 2. Утечка конфиденциальной информации через ИИ-чат-бот

Описание ситуации.

Сотрудник компании использует внешний генеративный ИИ-сервис для подготовки отчёта. Чтобы получить более точный результат, он загружает файл, содержащий фамилии клиентов, контактные данные и сведения о заключённых договорах. При этом сотрудник не проверяет правила обработки информации выбранным сервисом и внутренние требования организации.

Анализ проблемы.

Передача данных стороннему сервису может создать риск раскрытия персональной или коммерчески конфиденциальной информации. Допустимость такой передачи зависит от правового основания, условий

использования сервиса, договорных обязательств и применимых требований законодательства.

Возможные нарушения этических принципов:

1. нарушение конфиденциальности;
2. недостаточная ответственность при работе с данными;
3. отсутствие должной оценки рисков.

Возможные последствия:

1. раскрытие сведений о клиентах;
2. нарушение договорных обязательств;
3. финансовые и репутационные потери;
4. необходимость расследования инцидента и принятия предусмотренных законом мер.

Предлагаемые решения:

1. Запретить передачу конфиденциальных данных в неутверждённые ИИ-сервисы.
2. Использовать обезличенные или искусственно созданные данные, если они подходят для задачи.
3. Проверять условия хранения и обработки информации поставщиком ИИ-сервиса.
4. Разработать внутренние правила использования генеративного ИИ.
5. Обучить сотрудников безопасной работе с интеллектуальными инструментами.

Вывод по кейсу: перед использованием внешнего ИИ-сервиса необходимо оценивать риски передачи информации и соблюдать установленные правила защиты данных.

3. Кейс № 3. Ошибки системы распознавания лиц

Описание ситуации.

Организация использует систему распознавания лиц для контроля доступа в здание. В одном из случаев система ошибочно идентифицировала посетителя и отказала ему во входе. Сотрудник охраны полностью доверился результату автоматической проверки и не предложил альтернативный способ подтверждения личности.

Анализ проблемы.

Ошибки распознавания могут возникать из-за качества изображения, условий освещения, особенностей обучающих данных и ограничений алгоритма.

Проблема заключается не только в технической ошибке, но и в отсутствии процедуры её проверки. Автоматический результат используется без дополнительного контроля, хотя он непосредственно влияет на права и возможности человека.

Возможные последствия:

1. необоснованный отказ в доступе;
2. нарушение нормального рабочего процесса;
3. возникновение конфликтных ситуаций;
4. риск неравномерной точности распознавания для разных групп людей.

Предлагаемые решения:

1. Проверить точность системы в условиях реальной эксплуатации.
2. Предусмотреть альтернативный способ подтверждения личности.
3. Обеспечить участие сотрудника в рассмотрении спорных случаев.
4. Фиксировать ошибки и регулярно анализировать причины.
5. Проверить законность и соразмерность использования биометрических данных.

Вывод по кейсу: решения, влияющие на доступ человека к услугам или объектам, не должны безусловно

зависеть от одного автоматического результата. Необходимо предусматривать возможность проверки и исправления ошибок.

4. Итоговая таблица анализа кейсов

Кейс	**Основная этическая проблема**	**Возможные последствия**	**Предлагаемые меры**
Отбор кандидатов	Предвзятость алгоритма	Необоснованный отказ в трудоустройстве	Аудит данных и контроль решений
ИИ-чат-бот	Риск утечки данных	Раскрытие персональной или коммерческой информации	Ограничение доступа и проверка сервисов
Распознавание лиц	Ошибочная идентификация	Необоснованный отказ в доступе	Альтернативная проверка и мониторинг точности

5. Самостоятельная часть задания

Выберите ещё один кейс для анализа:

1. ИИ формирует неверные ответы в образовательном чат-боте.
2. Алгоритм кредитного скоринга необоснованно отклоняет заявки клиентов.
3. Генеративный ИИ создаёт изображение реального человека без его согласия.
4. Система мониторинга сотрудников оценивает их продуктивность без учёта контекста работы.
5. ИИ формирует ложные сведения о компании или человеке.

Для выбранного кейса необходимо описать ситуацию, выявить этические проблемы, определить возможные последствия и предложить не менее четырёх мер по снижению рисков.

Вывод

В ходе выполнения практической работы проведён анализ этических проблем использования искусственного интеллекта в информационных системах. Рассмотрены риски алгоритмической предвзятости, нарушения конфиденциальности и ошибок автоматической идентификации. Определены возможные последствия для пользователей и организаций, а также предложены меры по обеспечению справедливости, безопасности, прозрачности и ответственного применения ИИ.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.

– Оформление отчёта — 1 балл.

Практическая работа №16. Исследование правовых аспектов использования ИИ в бизнесе
Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.4. Контролировать результат обучения
ПК 3.5. Оформлять результат проведения процедуры обучения

Задание:

Рассмотрите ситуацию.

Компания «TechMarket» занимается продажей бытовой техники через интернет-магазин. Для повышения эффективности бизнеса руководство решило внедрить три ИИ-системы:

— ИИ-чат-бот для автоматического общения с покупателями и обработки обращений.

— Систему персональных рекомендаций для подбора товаров на основе истории просмотров и покупок пользователей.

— Систему анализа отзывов для определения настроений клиентов и выявления проблем с качеством обслуживания.

При внедрении этих систем возникли вопросы, связанные с обработкой персональных данных, достоверностью предоставляемой информации, использованием материалов из Интернета и ответственностью за ошибки ИИ.

Необходимо выполнить следующие задания.

1. Определить правовые риски.

Для каждой ИИ-системы укажите не менее двух возможных правовых рисков.

2. Подобрать нормативные правовые акты.

Определите, какие законодательные акты могут применяться к каждой ситуации, и объясните почему.

3. Предложить меры по снижению рисков.

Разработайте не менее трёх конкретных мер, которые помогут компании соблюдать законодательство при использовании ИИ.

4. Рассмотреть ситуацию с персональными данными.

Чат-бот запрашивает у покупателя фамилию, имя, номер телефона и адрес доставки. Определите, какие требования необходимо учитывать при сборе и обработке этих сведений.

5. Проанализировать авторские права.

Компания использует ИИ для создания рекламных изображений товаров. Часть изображений похожа на работы известных художников. Определите возможные правовые риски и предложите способы их

уменьшения.

6. Сформулировать вывод.

Объясните, почему внедрение ИИ в бизнес требует не только технического тестирования, но и правовой оценки.

Пример решения:

1. Анализ правовых рисков

ИИ-система	Возможные правовые риски
ИИ-чат-бот	Сбор персональных данных без надлежащего правового основания; предоставление покупателю недостоверной информации о товаре или условиях возврата
Система рекомендаций	Использование персональных данных без соблюдения требований законодательства; непрозрачная обработка сведений о поведении пользователей
Анализ отзывов	Незаконное использование персональных данных из отзывов; ошибочная классификация отзывов и принятие на её основе необоснованных решений

2. Нормативные правовые акты

Нормативный акт	Применение в ситуации
Федеральный закон № 152-ФЗ	Обработка имени, телефона, адреса и других персональных данных покупателей
Федеральный закон № 149-ФЗ	Защита информации и обеспечение безопасности информационных систем
Закон РФ № 2300-1	Предоставление покупателям достоверной информации о товарах, продавце и условиях покупки
Федеральный закон № 38-ФЗ	Проверка рекламных материалов и рекламных сообщений, созданных с использованием ИИ
Часть четвёртая ГК РФ	Оценка правомерности использования чужих изображений, текстов и других охраняемых результатов интеллектуальной деятельности

Вывод: компания должна оценивать правовые требования применительно к каждой системе. Например, чат-бот, обрабатывающий адрес доставки, требует особого внимания к персональным данным, а

генератор рекламных изображений — к авторским и исключительным правам.

3. Меры по снижению правовых рисков

Компания «TechMarket» может принять следующие меры:

1. Проверять правовые основания обработки данных. До внедрения ИИ определить цели обработки, состав данных, основания обработки и сроки хранения.
2. Ограничивать сбор информации. Запрашивать только те данные, которые необходимы для конкретной задачи.
3. Защищать информацию. Ограничивать доступ сотрудников к данным, использовать средства защиты и контролировать действия пользователей информационной системы.
4. Проверять ответы ИИ. Настроить контроль ответов чат-бота, особенно по вопросам оплаты, гарантий, возврата товаров и прав потребителей.
5. Проверять рекламные материалы. Убедиться, что изображения и тексты не нарушают права третьих лиц.
6. Назначить ответственных сотрудников. Определить, кто контролирует работу ИИ, рассматривает жалобы и исправляет выявленные ошибки.
7. Проводить регулярные проверки. Анализировать качество работы систем и актуальность внутренних правил с учётом изменений законодательства.

4. Обработка персональных данных покупателей

Чат-бот собирает фамилию, имя, номер телефона и адрес доставки. Эти сведения могут относиться к персональным данным, поскольку позволяют идентифицировать человека или связаны с ним.

При организации обработки компания должна учитывать следующие требования:

— Определить законное основание обработки персональных данных. Согласие является одним из возможных оснований, но не единственным: применимость других оснований зависит от конкретной ситуации.

— Чётко сформулировать цели обработки, например оформление и доставка заказа.

— Не собирать сведения, которые не нужны для достижения заявленной цели.

— Предоставить пользователю необходимую информацию об обработке данных в соответствии с требованиями законодательства.

— Обеспечить конфиденциальность и безопасность информации.

— Установить сроки хранения и порядок удаления данных после достижения целей обработки, если отсутствуют другие законные основания для хранения.

— Проверить, передаются ли данные сторонним поставщикам ИИ-сервисов и на каких условиях осуществляется такая передача.

Пример правильного подхода: для оформления доставки чат-бот может запросить имя получателя, телефон и адрес. При этом компании следует объяснить, для чего нужны эти сведения, определить основание обработки и обеспечить их защиту.

5. Авторские права на изображения, созданные ИИ

Если рекламные изображения похожи на произведения известных художников, необходимо оценить

возможные риски нарушения авторских прав.

К возможным рискам относятся:

— Использование охраняемых элементов чужого произведения без необходимого правового основания.

— Применение изображений, полученных из Интернета, без проверки условий их использования.

— Использование генератора изображений, условия которого ограничивают коммерческое применение результатов.

Для снижения рисков компании следует:

1. Проверять исходные материалы и условия использования выбранного ИИ-сервиса.
2. Не копировать намеренно охраняемые изображения и их оригинальные элементы без соответствующих прав.
3. Использовать собственные материалы, лицензированные изображения или контент с подходящими условиями использования.
4. Проверять готовые изображения перед публикацией.
5. При необходимости получать юридическую консультацию по конкретному материалу.

Важно учитывать, что само по себе сходство с художественным стилем не означает автоматически нарушения авторских прав. Правовая оценка зависит от конкретных обстоятельств, в том числе от того, воспроизводятся ли охраняемые элементы произведения.

6. Итоговый вывод

Использование искусственного интеллекта позволяет компаниям автоматизировать бизнес-процессы, ускорить обслуживание клиентов и повысить эффективность работы. Однако внедрение ИИ может сопровождаться правовыми рисками, связанными с обработкой персональных данных, предоставлением информации потребителям, рекламой и интеллектуальной собственностью.

Поэтому перед внедрением ИИ необходимо не только проверить техническую работоспособность системы, но и определить применимые требования законодательства, установить правила обработки данных, организовать контроль результатов и назначить ответственных сотрудников.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №17. Анализ рисков использования ИИ в информационных системах
Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.4. Контролировать результат обучения
ПК 3.5. Оформлять результат проведения процедуры обучения

Задание:

Компания «SmartOffice» внедрила информационную систему с использованием искусственного интеллекта. Система предназначена для автоматизации работы отдела кадров, службы поддержки и отдела продаж.

В состав системы входят следующие модули:

— ИИ-модуль анализа резюме кандидатов на вакансии.

— ИИ-чат-бот для общения с клиентами.

— ИИ-модуль прогнозирования продаж на основе исторических данных.

— ИИ-модуль формирования отчётов по внутренним документам компании.

После запуска системы руководство обнаружило несколько проблем:

1. ИИ-модуль анализа резюме отклоняет большое количество кандидатов определённого возраста, хотя их квалификация соответствует требованиям вакансии.
2. Чат-бот иногда сообщает клиентам неверные сведения о стоимости товаров и условиях возврата.
3. В прогнозах продаж возникают значительные ошибки из-за использования устаревших данных.
4. Сотрудник загрузил во внешний ИИ-сервис документ, содержащий конфиденциальную информацию о клиентах.
5. После обновления модели время обработки запросов увеличилось, а некоторые функции системы перестали работать корректно.

Необходимо выполнить следующие задания.

1. Выявить риски.

Определите тип риска для каждой из пяти ситуаций. Объясните, в чём заключается проблема и какие последствия она может вызвать.

2. Оценить уровень рисков.

Для каждой ситуации определите вероятность возникновения риска и тяжесть последствий по шкале от 1 до 5.

Рассчитайте уровень риска по формуле:

R=P \times S

Пример решения:

1. Выявление рисков

№	Ситуация	Тип риска	Возможные последствия
1	Предвзятый анализ резюме	Этический, правовой	Необоснованный отказ кандидатам, дискриминация, жалобы

2	Недостоверные ответы чат-бота	Технический, правовой, репутационный	Жалобы клиентов, финансовые потери, нарушение прав потребителей
3	Использование устаревших данных	Риск качества данных	Ошибочные прогнозы, неправильное планирование закупок
4	Передача конфиденциальных сведений внешнему сервису	Информационная безопасность, правовой	Утечка данных, нарушение обязательств по конфиденциальности
5	Сбой после обновления модели	Технический, организационный	Простой системы, задержки обработки запросов, потеря производительности

2. Оценка уровня рисков

Пример заполнения таблицы:

№	Тип риска	Вероятность	Тяжесть	Расчёт	Уровень
1	Этический, правовой	4	5		Критический
2	Технический, правовой	4	4		Критический
3	Качество данных	4	3		Высокий
4	Информационная	3	5		Высокий

	безопасность				
5	Технический	3	4		Высокий

Оценки в таблице являются примером. При выполнении работы студент может предложить другие значения, если аргументирует свою оценку.

Пример расчёта для ситуации № 1:

Вероятность возникновения предвзятого отбора оценивается в 4 балла, поскольку проблема уже обнаружена. Тяжесть последствий составляет 5 баллов, так как решения могут существенно повлиять на трудоустройство людей и создать правовые риски.

$R=4 \times 5=20$

По принятой шкале это критический риск, требующий первоочередного внимания.

3. Приоритеты устранения рисков

В первую очередь следует обратить внимание на следующие проблемы:

1. Предвзятый анализ резюме. Необходимо проверить работу модели, выявить причины необоснованных отказов и временно ограничить автоматическое отклонение кандидатов до устранения проблемы.

2. Недостоверные ответы чат-бота. Следует проверить ответы по стоимости товаров, гарантиям и возвратам. До исправления системы спорные запросы необходимо передавать сотрудникам.

3. Передача конфиденциальных данных. Необходимо прекратить неразрешённую передачу информации, оценить последствия инцидента и принять меры в соответствии с внутренними процедурами и требованиями законодательства.

Эти проблемы имеют высокий потенциальный ущерб для людей, клиентов и самой компании. При этом конкретная очерёдность реагирования должна учитывать фактический масштаб инцидента: например, подтверждённая утечка чувствительных данных может потребовать немедленных действий независимо от числового рейтинга.

4. Меры по снижению рисков

№	Риск	Меры предупреждения и снижения
1	Предвзятый анализ резюме	Проверять данные и результаты модели на наличие дискриминационных закономерностей; предусмотреть человеческую проверку решений; регулярно тестировать систему на

		разных группах кандидатов
2	Недостоверные ответы чат-бота	Использовать проверенную базу знаний; ограничить ответы в областях с повышенными последствиями ошибок; передавать спорные вопросы сотруднику
3	Устаревшие данные	Регулярно обновлять наборы данных; проверять их полноту и актуальность; контролировать точность прогнозов на новых данных
4	Утечка конфиденциальной информации	Запретить передачу конфиденциальных документов в неразрешённые сервисы; разграничить права доступа; обучить сотрудников правилам информационной безопасности
5	Сбой после обновления	Тестировать обновления в отдельной среде; сохранять предыдущую стабильную версию; внедрять мониторинг производительности и механизм отката

5. План управления рисками

Риск	**Предупреждение**	**Обнаружение**	**Действия при проблеме**	**Ответственный**
Предвзятый отбор кандидатов	Проверка алгоритма и данных	Анализ статистики решений	Приостановить автоматический отбор, проверить модель	HR-специалист и разработчик

Ошибки чат-бота	Проверенная база знаний	Анализ жалоб и тестирование ответов	Исправить базу знаний, передать спорные обращения сотрудникам	Руководитель службы поддержки
Устаревшие данные	Регламент обновления данных	Контроль качества данных и точности прогнозов	Обновить данные и пересчитать прогнозы	Аналитик данных
Утечка информации	Контроль доступа и правила использования сервисов	Анализ журналов событий	Ограничить доступ, оценить инцидент, выполнить необходимые процедуры реагирования	Специалист по информационной безопасности
Сбой после обновления	Тестирование и резервная версия	Мониторинг ошибок и производительности	Откатить обновление, восстановить работоспособность	Системный администратор и разработчик

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

Практическая работа №18. Разработка отчёта по соблюдению законодательства при внедрении ИИ
Проверяемые компетенции:
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках
ПК 3.4. Контролировать результат обучения
ПК 3.5. Оформлять результат проведения процедуры обучения

Задание:

Компания «DigitalService» внедрила информационную систему с использованием искусственного интеллекта для автоматизации обслуживания клиентов.

Система включает три модуля:

— ИИ-чат-бот, который отвечает на вопросы клиентов и принимает обращения.

— ИИ-модуль анализа клиентских данных, который формирует рекомендации и прогнозирует потребности покупателей.

— ИИ-модуль генерации рекламных текстов и изображений.

В ходе внутренней проверки были выявлены следующие обстоятельства:

1. Чат-бот собирает имена, номера телефонов и адреса электронной почты клиентов, однако на сайте отсутствует понятное описание целей обработки данных.
2. Сотрудники используют внешний ИИ-сервис для анализа клиентских обращений, в том числе содержащих персональные данные.
3. ИИ иногда формирует рекламные тексты с неподтверждёнными сведениями о свойствах товаров.
4. Компания не проверяет, не нарушают ли сгенерированные изображения права третьих лиц.
5. В организации отсутствует документ, определяющий порядок контроля результатов работы ИИ и реагирования на ошибки.

На основании исходных данных необходимо разработать отчёт о соблюдении законодательства при внедрении ИИ.

Выполните следующие задания.

1. Описать информационную систему.

Укажите назначение системы, её основные модули, категории обрабатываемых данных и возможные последствия ошибок ИИ.

2. Определить применимые нормативные правовые акты.

Для каждой выявленной проблемы укажите законодательный акт, который необходимо учитывать, и объясните его применимость.

3. Провести анализ соблюдения законодательства.

Для каждой ситуации определите статус проверки:

— соответствует требованиям;

— выявлено несоответствие;

— требуется дополнительная проверка.

Обоснуйте каждый выбранный статус. Не делайте окончательный вывод о нарушении закона только на основании отсутствия документа или отдельных сведений, если для этого недостаточно фактов.

4. Составить таблицу выявленных проблем.

Включите в неё следующие сведения:

— номер проблемы;

— описание ситуации;

— применимые нормативные правовые акты;

— выявленный риск или несоответствие;

— возможные последствия;

— рекомендуемые меры.

5. Разработать план корректирующих мероприятий.

Для каждой проблемы определите конкретное действие, ответственного сотрудника, срок выполнения и ожидаемый результат.

6. Подготовить итоговый отчёт.

Оформите результаты проверки в виде документа со следующей структурой:

— наименование организации;

— название отчёта;

— дата составления;

— цель проверки;

— описание информационной системы;

— нормативная база;

— результаты проверки;

— выявленные риски и несоответствия;

— план корректирующих мероприятий;

— заключение.

Пример решения:

1. Описание информационной системы

Информационная система ООО «DigitalService» предназначена для автоматизации взаимодействия с клиентами и поддержки маркетинговой деятельности.

В системе используются три ИИ-модуля:

— чат-бот для обработки обращений клиентов;

— модуль анализа клиентских данных и формирования рекомендаций;

— модуль генерации рекламных текстов и изображений.

Система может обрабатывать персональные данные клиентов, сведения об их обращениях, информацию о товарах и рекламные материалы. Ошибки в работе модулей могут привести к утечке данных, распространению недостоверной информации, нарушению интеллектуальных прав и снижению доверия клиентов.

2. Нормативная база

Нормативный акт	Область применения
Федеральный закон № 152-ФЗ	Обработка и защита персональных данных
Федеральный закон № 149-ФЗ	Защита информации и правовые основы информационных технологий
Закон РФ № 2300-1	Достоверность информации, предоставляемой потребителям
Федеральный закон № 38-ФЗ	Соблюдение требований к рекламе
Часть четвёртая ГК РФ	Авторские и иные интеллектуальные права

При проведении полноценной проверки необходимо изучить актуальные редакции применимых норм, внутренние документы организации, договоры с поставщиками ИИ-сервисов и фактические процессы обработки данных.

3. Результаты проверки

№	Выявленная ситуация	Статус проверки	Обоснование
1	На сайте отсутствует понятное описание целей обработки данных	Выявлено несоответствие, требующее устранения	Необходимо проверить выполнение требований к информированию субъектов персональных данных и наличие соответствующих документов
2	Персональные данные передаются внешнему ИИ-сервису	Требуется дополнительная проверка	Необходимо установить правовое основание передачи, условия договора, состав передаваемых данных и меры

			защиты
3	ИИ формирует рекламные тексты с неподтверждённым и утверждениями	Выявлено несоответствие внутренним требованиям к проверке рекламы; правовая оценка требует проверки конкретных материалов	Необходимо проверить достоверность утверждений и соответствие требованиям законодательства о рекламе и защите потребителей
4	Не проверяются права на сгенерированные изображения	Требуется дополнительная проверка	Отсутствие проверки создаёт риск нарушения интеллектуальных прав, но само по себе ещё не доказывает факт нарушения
5	Нет документа о контроле ИИ и реагировании на ошибки	Выявлено организационное несоответствие	Отсутствует установленный внутренний порядок контроля, что повышает вероятность повторения ошибок

4. Анализ проблем и рекомендации

№	Проблема и возможные последствия	Рекомендуемые меры
1	Недостаточная прозрачность обработки персональных данных. Возможны жалобы клиентов и претензии контролирующих органов	Определить цели и правовые основания обработки, подготовить необходимые документы и обеспечить доступность предусмотренной законом информации
2	Передача данных внешнему сервису. Возможна утечка информации или нарушение требований к обработке	Проверить условия использования сервиса, минимизировать передаваемые данные,

	персональных данных	ограничить доступ и оценить законность передачи
3	Недостоверная реклама. Возможны введение потребителей в заблуждение, жалобы и юридическая ответственность	Ввести обязательную проверку рекламных утверждений сотрудником до публикации
4	Неизвестный статус прав на изображения. Возможны претензии правообладателей и необходимость удаления материалов	Проверять происхождение изображений, условия лицензий и риски использования охраняемых элементов
5	Отсутствие контроля ИИ. Ошибки могут оставаться незамеченными и повторяться	Утвердить регламент контроля, назначить ответственных и организовать регистрацию инцидентов

5. План корректирующих мероприятий

№	Мероприятие	Ответственный	Срок	Ожидаемый результат
1	Проверить процессы обработки персональных данных и актуализировать документы	Ответственный за обработку персональных данных, юрист	5 рабочих дней	Установлены цели, основания и правила обработки
2	Провести аудит внешнего ИИ-сервиса и передаваемых ему данных	Специалист по информационной безопасности, юрист	7 рабочих дней	Определены допустимый состав данных и условия передачи

3	Ввести проверку рекламных материалов перед публикацией	Руководитель маркетинга	3 рабочих дня	Исключена публикация непроверенных утверждений без согласования
4	Разработать процедуру проверки прав на изображения	Юрист, маркетолог	5 рабочих дней	Установлены правила проверки и подтверждения прав на материалы
5	Утвердить регламент контроля ИИ и реагирования на инциденты	Руководитель организации, ИТ-специалист	10 рабочих дней	Назначены ответственные и установлен порядок действий при ошибках

Указанные сроки являются примером для учебной работы. В реальной организации сроки определяются с учётом тяжести рисков и фактических обстоятельств.

6. Итоговое заключение

По результатам предварительной проверки установлено, что внедрение ИИ в ООО «DigitalService» сопровождается правовыми и организационными рисками. Наиболее существенное внимание необходимо уделить обработке персональных данных, передаче информации внешним сервисам и проверке рекламных материалов.

Для повышения соответствия требованиям законодательства рекомендуется провести полноценную проверку процессов обработки данных, пересмотреть условия использования внешних ИИ-сервисов, организовать контроль генерируемого контента и утвердить внутренний регламент эксплуатации ИИ.

Общий вывод: на основании представленных сведений нельзя подтвердить полное соответствие системы законодательству. До завершения дополнительных проверок и выполнения корректирующих мероприятий эксплуатацию отдельных функций следует оценивать с учётом уровня выявленных рисков.

Критерии оценки практических заданий:

- Соответствие требованиям задачи — 1 балл.
- Корректность и работоспособность кода / тестов — 1 балл.
- Качество тестового покрытия и полнота проверок — 1 балл.
- Применение инструментов тестирования и анализа — 1 балл.
- Оформление отчёта — 1 балл.

3.3. Задания для оценки освоения МДК 03.03

Раздел 1. Основы промпт-инжиниринга.

Тема 1.1 Введение в промпт-инжиниринг.

Практическое занятие №1. Формулировка цели задачи и создания простого промпта; анализ ответов модели.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта*
- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*

Цель:

1. познакомиться с понятием промпта и его классификацией;
2. научиться формулировать простые эффективные промпты для генерации текста;
3. научиться анализировать качество вывода по критериям структуры промпта и требований к выводу.

Задание

- Разработать 3 простых промпта для задач генерации текста (каждый промпт — для своей задачи):
    - Промпт 1: zero-shot формат, задача — краткое резюме по теме.

    - Промпт 2: один пример (one-shot) внутри промпта — задача по тем же темам.
    - Промпт 3: несколько примеров (few-shot) внутри промпта — задача по тематике сборника основных идей.
- Для каждого промпта кратко указать:
    - цель задачи и желаемый формат вывода (например, список из пунктов, краткое резюме, план статьи);
    - ограничение по объему и стилю;
    - возможные варианты анализа результата: полнота, точность, структура, стиль, нейтральность.
- После каждого промпта запишите:
    - Что получилось хорошо: соответствие формату, ясность, полнота.
    - Что можно улучшить: точность, глубина, стиль, соблюдение ограничений.
    - Влияние объема контекста на вывод: длинный текст — лучше или хуже учитывать.
    - Влияние типа промпта на качество вывода (zero-shot vs few-shot и т. д.).
- Составьте таблицу с промптом, типом промпта, выводом и кратким анализом.

Пример решения

№	Промпт	Тип промпта	Вывод модели	Краткий анализ
1	Напиши краткое резюме по следующему тексту: Искусственный интеллект развивается быстрыми темпами: от ранних концепций к	Zero-shot	Резюме: Искусственный интеллект развивался от ранних концепций к современным трансформерам. Современные подходы включают самообучение и RLHF. Применения охватывают	Плюсы: кратко и нейтрально; соблюд формальные требования вывода. Минусы: не раскрыты конкретные технологии; можно добавить примеры

	современным языковым моделям. Современные подходы включают трансформеры, самообучение и RLHF. Применения — медицина, финансы и образование; важные вызовы — этика и приватность.		медицину, финансы и образование; важные вызовы — этика и приватность.	технологий и ограничение по объёму.
2	Ввод/Вывод внутри промпта: Пример 1: Ввод: "Искусственный интеллект становится мощной технологией анализа больших данных и автоматизации задач в бизнесе."; Вывод: "Искусственный интеллект становится мощной технологией	One-shot	Искомое резюме: Искусственный интеллект эволюционировал от концепций к современным трансформерам; используются методы самообучения и RLHF. Применения в медицине, финансах и образовании; этика и приватность остаются ключевыми вызовами.	Преимущество: наличие примера внутри промпта повышает структурность вывода и соответствие формату. Недостаток: требует аккуратности с копированием примера; риск несоответствия тексту.

	анализа больших данных и автоматизации бизнес-процессов". Задача: На входе ниже текст: "Современные подходы включают трансформеры, обучение с подкреплением и RLHF; примеры применения — медицина, финансы, образование". Сгенерируй резюме в формате: краткое резюме (до 4–5 предложений).			
3	Вставьте 2–3 пары “Ввод/Вывод” (пример резюме). Затем: На входе ниже текст "Современные подходы включают трансформеры, RLHF и	Few-shot	Резюме: Эволюция ИИ ведёт к трансформерам и большим языковым моделям; обучение на больших корпусах и RLHF улучшают качество вывода. Применения — медицина, финансы, образование; вызовы	Few-shot обычно даёт более точный стиль и формат вывода; требует большего объёма внутри промпта. Возможны коррекции формулировок под конкретный текст.

	самообучение; применения — медицина, финансы, образование; вызовы — этика, приватность, безопасность". Сгенерируй резюме в формате из примеров (3–4 предложения).		— этика, приватность, безопасность.	

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

Практическое занятие №2. Сравнение инструктивного промпта и примера промпта на одной задаче; запись выводов.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей*

искусственного интеллекта

- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*

Цель

Научиться на практике видеть разницу между двумя фундаментальными техниками составления промптов:

- **Инструктивный промпт** *(Instructional Prompting)*: когда мы даём нейросети чёткую команду, объясняем, что нужно сделать.
- **Промпт-пример** *(Example-based Prompting / Few-shot)*: когда мы не даём прямой инструкции, а показываем один или несколько примеров желаемого результата, чтобы модель сама поняла паттерн.

Задача — понять, в каких ситуациях какой подход работает лучше, и научиться формулировать выводы на основе полученного опыта.

Задание

Шаг 1. Выбор задачи
Выберите одну из следующих задач (или придумайте свою аналогичную):

- **Задача А (Классификация):** определить эмоциональную окраску (позитивная, негативная, нейтральная) короткого отзыва.
- **Задача Б (Трансформация):** переписать официальное, канцелярское предложение в простой и понятный язык.
- **Задача В (Творчество):** сгенерировать слоган для кофейни, которая использует только свежеобжаренные зёрна из Эфиопии.

Шаг 2. Создание инструктивного промпта
Сформулируйте промпт, который прямо *инструктирует* модель выполнить задачу.

- **Ключевые элементы:** глагол действия («классифицируй», «перепиши», «придумай»), чёткое описание входных данных и ожидаемого формата вывода.

- **Пример для задачи А:** «Проанализируй следующий текст отзыва и определи его тональность. Варианты тональности: "позитивная", "негативная", "нейтральная". В ответе укажи только выбранный вариант. Текст отзыва: "Кофе был неплохим, но обслуживание оставило желать лучшего"».

Шаг 3. Создание промпта-примера
Сформулируйте промпт, который *не содержит прямой инструкции*, а показывает модели пример.

- **Ключевые элементы:** один или несколько примеров в формате «Вход → Выход». Важно, чтобы пример был релевантен задаче. Сама задача для модели должна быть сформулирована как новый, «голый» пример, который ей нужно завершить.
- **Пример для задачи А:**
«Вот несколько примеров:
- "Это лучшее, что я пробовал за последнее время!" → позитивная
- "Продукт пришёл с браком, деньги на ветер." → негативная
- "Товар соответствует описанию, доставка в срок." → нейтральная

Теперь определи тональность для этого отзыва: "Кофе был неплохим, но обслуживание оставило желать лучшего"».

Шаг 4. Выполнение и сравнение
Отправьте оба промпта (из Шага 2 и Шага 3) нейросети. Сравните полученные ответы по следующим критериям:

- **Точность:** какой ответ более корректен с точки зрения здравого смысла?
- **Формат:** какой ответ точнее соответствует желаемому формату (например, только одно слово без лишних пояснений)?
- **Стиль:** какой ответ больше соответствует неявным ожиданиям (например, более творческий или, наоборот, более сухой)?
- **Предсказуемость:** какой подход дал более стабильный и ожидаемый результат?

Шаг 5. Запись выводов
Оформите свои наблюдения в виде краткого аналитического текста (3–5

предложений). В выводах укажите:

- Какой подход (инструкция или пример) сработал лучше для вашей конкретной задачи и почему.
- В каких ситуациях, по вашему мнению, стоит использовать инструктивный промпт, а в каких — промпт-пример.
- Что было самым сложным при составлении каждого из промптов.

Пример решения

Выбранная задача: Задача Б (трансформация канцелярского стиля).

Инструктивный промпт:

Перепиши следующий текст, написанный сухим официальным языком, в простой и понятный для обычного человека вид. Сохрани основной смысл, но избавься от канцеляризмов. Результат должен состоять из одного предложения.
Исходный текст: «В целях оптимизации логистических процессов и сокращения временных издержек, доставка вашего заказа будет произведена в течение временного интервала с 14:00 до 18:00».

Ответ нейросети на инструктивный промпт:

Чтобы сэкономить время и оптимизировать доставку, ваш заказ привезут сегодня с 14:00 до 18:00.

Промпт-пример:

Переведи с "канцелярского" на человеческий:

Исходный текст: «Ввиду наличия неблагоприятных метеорологических явлений,

проведение мероприятий на открытом воздухе является нецелесообразным».

Простой текст: «Из-за плохой погоды мероприятие на улице проводить не будут».

Теперь переведи этот текст:

Исходный текст: «В целях оптимизации логистических процессов и сокращения временных издержек, доставка вашего заказа будет произведена в течение временного интервала с 14:00 до 18:00».

Ответ нейросети на промпт-пример:

Чтобы доставить заказ быстрее и сэкономить ресурсы, мы привезём его вам с 14:00 до 18:00.

Выводы (пример записи):
Для задачи по упрощению текста лучше сработал инструктивный промпт. Ответ получился более естественным и лаконичным. Подход с примером тоже дал хороший результат, но нейросеть излишне сфокусировалась на «экономии ресурсов», что было в примере, хотя в исходном тексте речь шла о «сокращении издержек» в целом. Это показывает главный недостаток промпта-примера: модель может слишком буквально копировать семантику и стиль примера, что снижает гибкость. Инструктивный подход здесь предпочтительнее, так как он даёт чёткую цель. В то же время, промпт-пример был проще в составлении — не нужно было подбирать правильные глаголы-инструкции. Примеры буду использовать для творческих задач, где нужно задать стилистический паттерн, а инструкции — для задач, где важны точность и формат.

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.

- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

Тема 1. 2 Базовые понятия.

Практическое занятие №3. Создание простых промптов и их анализ. Использование контекста и условий для улучшения ответов.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта*
- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*

Цель:

1. познакомиться с понятием промпта и его классификацией;
2. научиться формулировать простые эффективные промпты для генерации текста;
3. научиться анализировать качество вывода по критериям структуры промпта и требований к выводу.

Задание

- Разработать 3 простых промпта для задач генерации текста (каждый промпт — для своей задачи):
    - Промпт 1: zero-shot формат, задача — краткое резюме по теме.

    - Промпт 2: один пример (one-shot) внутри промпта — задача по тем же темам.
    - Промпт 3: несколько примеров (few-shot) внутри промпта — задача по тематике сборника основных идей.
- Для каждого промпта кратко указать:
    - цель задачи и желаемый формат вывода (например, список из пунктов, краткое резюме, план статьи);
    - ограничение по объему и стилю;
    - возможные варианты анализа результата: полнота, точность, структура, стиль, нейтральность.
- После каждого промпта запишите:
    - Что получилось хорошо: соответствие формату, ясность, полнота.
    - Что можно улучшить: точность, глубина, стиль, соблюдение ограничений.
    - Влияние объема контекста на вывод: длинный текст — лучше или хуже учитывать.
    - Влияние типа промпта на качество вывода (zero-shot vs few-shot и т. д.).
- Составьте таблицу с промптом, типом промпта, выводом и кратким анализом.

Пример решения

№	Промпт	Тип промпта	Вывод модели	Краткий анализ
1	Напиши краткое резюме по следующему тексту: Искусственный интеллект развивается быстрыми темпами: от ранних концепций к	Zero-shot	Резюме: Искусственный интеллект развивался от ранних концепций к современным трансформерам. Современные подходы включают самообучение и RLHF. Применения охватывают	Плюсы: кратко и нейтрально; соблюд формальные требования вывода. Минусы: не раскрыты конкретные технологии; можно добавить примеры

	современным языковым моделям. Современные подходы включают трансформеры, самообучение и RLHF. Применения — медицина, финансы и образование; важные вызовы — этика и приватность.		медицину, финансы и образование; важные вызовы — этика и приватность.	технологий и ограничение по объёму.
2	Ввод/Вывод внутри промпта: Пример 1: Ввод: "Искусственный интеллект становится мощной технологией анализа больших данных и автоматизации задач в бизнесе."; Вывод: "Искусственный интеллект становится мощной технологией	One-shot	Искомое резюме: Искусственный интеллект эволюционировал от концепций к современным трансформерам; используются методы самообучения и RLHF. Применения в медицине, финансах и образовании; этика и приватность остаются ключевыми вызовами.	Преимущество: наличие примера внутри промпта повышает структурность вывода и соответствие формату. Недостаток: требует аккуратности с копированием примера; риск несоответствия тексту.

	анализа больших данных и автоматизации бизнес-процессов". Задача: На входе ниже текст: "Современные подходы включают трансформеры, обучение с подкреплением и RLHF; примеры применения — медицина, финансы, образование". Сгенерируй резюме в формате: краткое резюме (до 4–5 предложений).			
3	Вставьте 2–3 пары “Ввод/Вывод” (пример резюме). Затем: На входе ниже текст "Современные подходы включают трансформеры, RLHF и	Few-shot	Резюме: Эволюция ИИ ведёт к трансформерам и большим языковым моделям; обучение на больших корпусах и RLHF улучшают качество вывода. Применения — медицина, финансы, образование; вызовы	Few-shot обычно даёт более точный стиль и формат вывода; требует большего объёма внутри промпта. Возможны коррекции формулировок под конкретный текст.

	самообучение; применения — медицина, финансы, образование; вызовы — этика, приватность, безопасность". Сгенерируй резюме в формате из примеров (3–4 предложения).		— этика, приватность, безопасность.	

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

Практическое занятие №4. Создание сценариев с использованием нескольких шагов.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей*

искусственного интеллекта

- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*

Цель

1. Освоить методы проектирования многошаговых промптов для текст-воздействия на генерацию изображений.
2. На практике применить парные, пошаговые и цепочные промпты, а также организовать многократные взаимодействия с моделью.
3. Научиться документировать процесс и ориентироваться на конкретные требования к результату.

Задание 1. Парные промпты

Цель: создать пару входного описания и готового кода/промпта для генерации изображения.

Задача: сформулировать входной текст (Input) и соответствующий выходной промпт (Output) для генерации изображения по заданной сцене.

Пошаговое решение

- Определите тему и желаемый стиль изображения.
- Сформулируйте ясное и сжатое Input-описание сцены.
- На основе Input составьте Output-промпт, который включает:
    - стиль и жанр;
    - композицию и ракурс;
    - палитру и освещение;
    - разрешение/соотношение сторон (если применимо);
    - любые запреты (negative prompts) или детали, которых следует избегать.
- Включите в Output примеры параметров, которые часто используют модели (пример: aspect ratio, quality, seeds, lighting, texture details).
- Убедитесь, что Output остаётся в рамках исходного Input и не добавляет лишних элементов.

Пример решения

1. Input (Описание сцены): Город будущего ночью, неоновые вывески зелёного цвета, мокрый асфальт после дождя, главный объект — электромобиль в центре кадра; стиль — киберпанк/реализм.
2. Output (промпт для генерации): Город будущего ночью с неоновыми зелёными вывесками, мокрый асфальт после дождя, электромобиль в центре кадра; стиль: киберпанк, фотореалистичная визуализация; ракурс: взгляд на город с уровня глаз; композиция: центральное положение главного объекта; освещение: неоновое, отражения на мокрой поверхности; палитра: голубовато-серая с акцентами зелёного; соотношение сторон: 16:9; детали:



высокая детализация, реалистичная фактура; Негативный промпт: низкое разрешение, мультяшность, размытость.

Задание 2. Пошаговые промпты

Цель: спроектировать пошаговый промпт, который постепенно развивает идею сцены до готового изображения.

Задача: разработать последовательность из 4–5 шагов, каждый шаг формирует часть изображения, накапливая детали.

Пошаговое решение

- Определите базовую концепцию.
- На каждом шаге добавляйте конкретику: композиция → стиль → освещение

→ палитра → финальные детали.

- Объединяйте результаты шага за шагом, показывая, как каждый шаг уточняет финальный промпт.
- В конце сформируйте итоговый финальный промпт, который можно отправить в модель.

Пример решения

Тема: Постер на тему транспорта в городе будущего.

1. Шаг 1 (концепт): "Город будущего с экологичными транспортами, минималистический стиль."
   Промпт шага 1: "Концепт минималистичного города будущего с экологичным транспортом, плавные линии, широкая панорама, чистая



   архитектура, мягкое утреннее освещение, соотношение сторон 16:9."

2. Шаг 2 (композиция): "Фокус на электромобиле на переднем плане, фон — городская панорама."
   Промпт шага 2: "Добавьте на передний план электромобиль, фон —

панорама города; правило третей; 16:9; мягкое боковое освещение."



3. Шаг 3 (стиль и детали): "Минимализм с акцентами зелёного, плоские тени, чистые линии."
   Промпт шага 3: "Стиль: минимализм с зелёными акцентами, плоские тени, чистые линии; палитра: бирюзовый, зелёный и серый; детализация: высокая,



   без перегрузки."

4. Шаг 4 (освещение и атмосфера): "Рассвет, тёплый свет, отражения на мокром асфальте."
   Промпт шага 4: "Освещение: рассвет, тёплый свет, отражения на мокром асфальте, атмосфера: спокойная и вдохновляющая; 16:9."



5. Шаг 5 (финальный промпт): "Собрать все элементы в один финальный промпт."

Промпт шага 5: "Минималистичный город будущего на рассвете; передний план: электромобиль; фон: панорама; палитра: бирюзово-зелёная и серые тона; освещение: тёплый рассвет; композиция: 16:9; детализация: высокая;



стиль: фотореализм."

Задание 3. Цепочные промпты

Цель: построить цепочку промптов, где каждый следующий промпт строится на выводе предыдущего шага (цепочка уточнений).

Задача: разработать три этапа цепочного промпта.

Пошаговое решение

- Определите базу: общая концепция изображения.
- На следующем шаге добавляйте детали по стилю и композиции, исходя из первого вывода.
- На третьем шаге добавляйте финальные детали и ограничения (разрешение, формат).

Пример решения

Тематика: Постер на тему "Транспорт в городе будущего".

6. Этап 0 (базовый промпт): "Город будущего с экологическим транспортом, минимализм." Цель: получить базовую визуальную концепцию. Промпт этапа 0: "Концепт: минималистичный город будущего с экологическим

транспортом, чистые линии, формат 16:9."

7. Этап 1 (детализация): "Добавь электробусы, велосипеды, зелёные акценты,





неоновые детали." Промпт этапа 1: "Минималистичный город будущего с электробусами и велосипедами, зелёные акценты, неоновые детали, чистые линии, формат 16:9."

8. Этап 2 (финализация): "Уточни освещение рассвета, мокрый асфальт, рефлекции, палитра: синий + зелёный" Фрагмент промпта: "Промпт этапа 2:



Минималистичный город на рассвете; главный объект в переднем плане — электромобиль; фон — городская панорама; освещение: тёплый рассвет, отражения на мокром асфальте; палитра: синий/зелёный; стиль: реализм; формат: 16:9."

Задание 4. Многократные взаимодействия

Цель: спроектировать план из нескольких раундов взаимодействия с моделью/генератором для итеративной доработки изображения.

Задача: разработать схему из 3 раундов взаимодействия, включая что именно вы будете менять в каждом раунде и какие замечания давать модели.

Пошаговое решение

- Определите цель и ожидаемый результат после каждого раунда.
- В первом раунде сгенерируйте базовое изображение и кратко опишите, что можно улучшить.
- Во втором раунде скорректируйте промпт и добавьте конкретные исправления (композиция, стиль, освещение, палитра).
- В третьем раунде примените финальные уточнения и получите окончательный вариант.
- Зафиксируйте итоговый промпт и сделайте мини-отчет: что изменилось и почему.

Пример решения

Тема: Постер экологичного транспорта в городе будущего.

- Раунд 1. Базовый промпт: "Минималистичный город будущего с экологическим транспортом, мягкий утренний свет, голубо-зелёная палитра, формат16:9."
  Замечания: требуется больше деталей по композиции и палитре, добавить отражения на мокром асфальте.

- Раунд 2. Уточнение: "Минималистичный город на рассвете с электробусами и велосипедами на переднем плане, панорама города на заднем плане, палитра: синий и зелёный, мягкие тени, reflections."
  Замечания: добавить акцент на главный объект (электромобиль в кадре) и более чёткие тени.

- Раунд 3. Финальные корректировки: "Минималистичный город будущего на рассвете; передний план: электромобиль; фон: панорама; освещение: тёплый рассвет; палитра: синий, зелёный, серый; композиция: правило третей, 16:9; рендер: высокая детализация, фотореализм."

Итог: получен завершённый постер с нужным стилем и деталями.



Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

Практическое занятие №5. Разработка набора базовых промптов: системный, пользовательский, роли; формирование контракта

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном*

и иностранном языках.

- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта*
- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*

Цель:
Научиться проектировать и составлять многоуровневую систему промптов для создания предсказуемого и управляемого ИИ-ассистента. В ходе задания будут отработаны:

Системный промпт (System Prompt): определение «личности», глобальных правил, ограничений и зоны ответственности модели.
Промпт роли (Role Prompt): задание конкретного профессионального или социального амплуа для выполнения задачи.
Пользовательский промпт (User Prompt): формулировка конкретной, ситуативной задачи в рамках заданной роли и системы.
Контракт (Contract): создание явного соглашения о формате, структуре и правилах взаимодействия, которое гарантирует стабильный результат.

Задание:
Шаг 1. Определение сценария
Выберите один из сценариев для разработки ИИ-ассистента (или придумайте свой):

Сценарий А: ассистент для студента, помогающий готовиться к экзаменам.
Сценарий Б: ассистент-маркетолог для малого бизнеса, генерирующий идеи для соцсетей.
Сценарий В: ассистент-редактор, помогающий писателю улучшать тексты (проверка стиля, диалогов, ритма).

Шаг 2. Разработка системного промпта
Создайте системный промпт, который задаёт общие «правила игры» для вашего ассистента. Он должен включать:

Личность и тон: как ассистент общается (поддерживающе, строго, иронично, академично)?
Глобальные ограничения: чего он никогда не должен делать (например, «не давать прямых ответов на экзаменационные вопросы», «не использовать токсичную лексику»).
Основная цель: его глобальная миссия в рамках сценария.

Шаг 3. Разработка промпта роли
Сформулируйте промпт, который активирует нужную роль. Этот промпт будет подставляться перед каждой задачей, относящейся к этой роли. Он должен чётко определить:

Профессиональную позицию: («Ты — старший редактор издательства...»).
Ключевые компетенции: («Твоя специализация — вычитка художественной прозы...»).
Цель роли в контексте задачи: («Твоя задача — не переписывать текст за автора, а задавать точные вопросы, которые помогут ему улучшить собственный стиль»).

Шаг 4. Разработка пользовательского промпта
Напишите конкретный пользовательский промпт, который ставит ассистенту одну чёткую задачу в рамках выбранной роли.

Пример: «Проанализируй следующий диалог между двумя персонажами. Отметь, где речь звучит неестественно для XXI века, и предложи по 2–3 варианта более живой замены для каждой реплики».

Шаг 5. Формирование контракта
Создайте «контракт» — это часть системного или ролевого промпта, которая жёстко определяет формат ответа. Он должен быть сформулирован как незыблемое правило.

Пример контракта: «Твой ответ ВСЕГДА должен состоять из двух частей, разделённых горизонтальной чертой:Анализ: краткий список (3–5 пунктов) найденных стилистических проблем.
Варианты: для каждой неестественной реплики — список из предложенных замен

в формате Исходная реплика → Вариант 1 / Вариант 2».

Шаг 6. Сборка и тестирование
Соберите финальный промпт для нейросети, склеив части в логическом порядке:

Системный промпт.
Промпт роли.
Контракт.
(Пустая строка)
Пользовательский промпт с конкретной задачей.

Отправьте этот составной промпт вместе с текстовыми данными (например, с тем самым диалогом) в нейросеть и оцените, насколько точно модель выполнила инструкцию и соблюла формат контракта.

Пример решения:
Сценарий: В (ассистент-редактор для писателя).

1. Системный промпт

Ты — виртуальный ассистент, выступающий в роли старшего литературного редактора. Твоя главная цель — помогать авторам развивать их уникальный голос, а не переписывать их тексты под шаблон. Ты тактичен, конструктивен и всегда аргументируешь свои предложения. Ты не должен хвалить текст без повода, твоя задача — находить точки роста. Избегай сложной редакторской терминологии, говори на языке пользы для автора.

2. Промпт роли

Сейчас ты работаешь в роли специалиста по диалогам и развитию персонажей. Твоя экспертиза — делать речь героев живой, индивидуальной и работающей на сюжет. Ты не исправляешь орфографию и пунктуацию, твоя единственная фокусная область — КАЧЕСТВО РЕЧИ персонажей.

3. Контракт

ФОРМАТ ОТВЕТА ЯВЛЯЕТСЯ СТРОГИМ КОНТРАКТОМ. Любое отклонение от него недопустимо. Твой ответ должен быть оформлен в виде Markdown-списка и содержать ровно два раздела:

1. Проблемные зоны

Пункт 1: [краткое описание проблемы, например: "Речь профессора звучит как речь подростка"].
Пункт 2: [...]

2. Предложения
Для каждого проблемного пункта приведи 2 варианта замены.

Вариант А (минимальные правки): [сохраняет 90% смысла, но исправляет проблему].
Вариант В (стилистический): [более смелый вариант, меняющий оттенок фразы].

4. Пользовательский промпт

Проанализируй следующий диалог из научно-фантастического рассказа.

Персонаж 1 (Капитан звездолёта): «Мы входим в астероидное поле. Я полагаю, нам следует активировать дефлекторные щиты».
Персонаж 2 (Инженер): «Слушаюсь, капитан. Мною будет произведена активация защитных систем в данную минуту».

Примени контракт и предоставь анализ с предложениями.

Результат работы нейросети (ожидаемый):

1. Проблемные зоны

Речь Инженера слишком формальная, канцелярская и медленная для экстренной ситуации на мостике. Пассивный залог («Мною будет произведена активация») делает её безжизненной.
Слово «полагаю» в речи Капитана звучит неуверенно. Капитан должен отдавать чёткие приказы.

2. Предложения

Проблема: речь Инженера

Вариант А (минимальные правки): «Есть, капитан. Активирую щиты».
Вариант В (стилистический): «Принято. Вывожу дефлекторы на полную. Защита будет через пять секунд».

Проблема: речь Капитана

Вариант А (минимальные правки): «Мы входим в астероидное поле. Активировать дефлекторные щиты».
Вариант В (стилистический): «Астероидное поле по курсу. Всю энергию на щиты. Живо!»

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

Практическое занятие №6. Оценка совместимости промптов с двумя разными моделями и фиксирование различий.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта*
- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*

Цель: Научиться эмпирически оценивать, как одна и та же инструкция (промпт) интерпретируется и исполняется разными нейросетевыми моделями. Задача — уйти от предположений к фактам, научиться выявлять систематические различия в поведении моделей (стиль, креативность, следование формату, «галлюцинации») и фиксировать их для дальнейшего использования в проектах

Задание:
Шаг 1. Подготовка
Выберите две доступные вам нейросетевые модели с разным позиционированием или архитектурой. Например:

Модель А: условно «творческая и разговорная» (например, GPT-4o, Claude).
Модель Б: условно «аналитическая и точная» (например, Llama, Gemini).

Подготовьте один нетривиальный, составной промпт, который проверяет сразу несколько навыков модели. Промпт должен включать:

Задание роли.
Конкретную задачу с неочевидным решением.

Строгий формат вывода (контракт).
Требование к стилю.

Шаг 2. Создание эталонного промпта
Сформулируйте и сохраните промпт, который вы будете использовать для обеих моделей.

Пример промпта:

Роль: ты — опытный сценарист диалогов для видеоигр в жанре киберпанк.

Задача: напиши короткий диалог (5–7 реплик) между уличным торговцем имплантами и скептически настроенным клиентом. Торговец должен использовать агрессивный, навязчивый маркетинг, а клиент — отшучиваться и задавать неудобные технические вопросы.

Контракт (формат ответа):
Ваш ответ должен быть в формате JSON и содержать только два ключа:

atmosphere: строка из 1–2 предложений, описывающая общую атмосферу сцены (например, «Неоновый дождь, запах озона и жареного фастфуда»).
dialogue: массив строк, где каждая строка — это одна реплика, начинающаяся с имени говорящего и двоеточия (например, "Торговец: Эй, киберпанк, глаза не намозолил? У меня есть то, что тебе нужно.").

Стиль: избегай клише вроде «хромированный» и «мегакорпорация». Сделай мир грязным, реалистичным и приземлённым.

Шаг 3. Проведение эксперимента

Отправьте этот промпт Модели А. Скопируйте полный промпт и полученный ответ в рабочую таблицу или документ.
Отправьте ТОТ ЖЕ САМЫЙ промпт Модели Б. Скопируйте её ответ рядом с

первым.

Шаг 4. Анализ и фиксирование различий
Сравните два полученных ответа и заполните сравнительную таблицу по следующим критериям:

Критерий Модель А Модель Б
Соблюдение формата (JSON) Да/Нет, были ли ошибки? Да/Нет, были ли ошибки?
Количество реплик [Число] [Число]
Атмосфера (ключ atmosphere) [Текст из ответа] [Текст из ответа]
Характер торговца Насколько он навязчив? Приведите цитату. Насколько он навязчив? Приведите цитату.
Характер клиента Насколько он скептичен? Приведите цитату. Насколько он скептичен? Приведите цитату?
Соблюдение стиля (отсутствие клише) Были ли клише? Если да, какие? Были ли клише? Если да, какие?
«Галлюцинации» или нелогичности Были ли фактические или логические ошибки? Были ли фактические или логические ошибки?
Общее впечатление (1–10) [Оценка] [Оценка]

Шаг 5. Формулировка выводов
На основе таблицы напишите краткий аналитический отчёт (5–7 предложений), в котором ответьте на вопросы:

Какая модель лучше справилась с ролью и задачей? Почему?
Какая модель строже соблюдала формат (контракт)?
В чём заключались самые яркие стилистические различия?
Для каких задач вы бы выбрали Модель А, а для каких — Модель Б, исходя из этого теста?

Пример решения:
Промпт: (используется эталонный из задания)

Ответ Модели А (условно «творческой»):

{
  "atmosphere": "Под навесом из гниющего пластика, который едва спасает от кислотного дождя. В воздухе висит гул генераторов и запах озона.",
  "dialogue": [
    "Торговец: Новое лёгкое? Твоё старое кашляет, как дизель на мазуте. Ставлю «Рио-Браво 7», хрипеть перестанешь, даже если дышать пеплом.",
    "Клиент: «Рио-Браво»? У них же теплообменник горит после трёх недель. Гарантия есть или как обычно?",
    "Торговец: Гарантия — это для корпоратов, которые пьют дистиллят. А для тебя — цена со скидкой. От сердца отрываю!"
"Клиент: От сердца ты оторвёшь, когда оно из-за твоего «Рио-Браво» встанет. Какая у него потребляемая мощность? Мой старый реактор потянет?",
    "Торговец: Мощность... Слушай, ты дышать хочешь или ватты считать? Бери, пока я добрый. Завтра на 200 кредитов дороже будет.",
    "Клиент: Завтра я либо найду «Танцующий лист», либо твою палатку закроют санитарные дроны. Так что с мощностью?",
    "Торговец: ...Ладно. 40 ватт в пике. Доволен? Теперь плати."
  ]
}

Ответ Модели Б (условно «аналитической»):

{
  "atmosphere": "Узкая аллея между двумя небоскрёбами. Сверху капает техническая жидкость, смешиваясь с неоновыми отблесками. Постоянный фоновый шум вентиляционных систем.",
  "dialogue": [
    "Торговец: Кибернетическая печень, 4-я модель. Фильтрует любой синтетический алкоголь. Бери, пока не подорожало.",
    "Клиент: 4-я модель? А протокол совместимости с портами «Сайбер-Нексус» 2045 года у неё какой? А то знаю я вас.",
    "Торговец: Протокол... стандартный. Встанет как родная, не сомневайся.",
    "Клиент: «Стандартный». То есть мне потом с паяльником под ребром лежать?

Назови точную версию прошивки.",

"Торговец: Слушай, умник, ты берёшь или нет? Очередь стоит.",

"Клиент: Очередь из таких же, как я? Хорошо. Цена. И есть ли у неё защита от перегрева при 100% нагрузке более 10 минут?",

"Торговец: ...На 10 минут хватит. Может быть. Цена — 800. И это я вам ещё должен останусь."

]

}

Заполненная таблица (фрагмент):

Критерий	Модель А	Модель Б
Соблюдение формата (JSON)	Да, идеально	Да, идеально
Атмосфера	Более «грязная», сенсорная, с запахами и звуками.	Более функциональная, с акцентом на визуальные детали и геометрию.
Характер торговца	Агрессивный, манипулятивный («от сердца отрываю»).	Прагматичный, уставший («очередь стоит»).
Характер клиента	Технический скепсис, давит на гарантию и совместимость.	Технический скепсис, давит на прошивку и конкретные ТТХ (нагрузка, время).
Соблюдение стиля	Клише нет. Язык очень живой, уличный.	Клише нет. Язык более сухой, технический, но соответствует ролям.
Общее впечатление (1–10)	9/10 (очень живо и кинематографично).	8/10 (логично и правдоподобно, но чуть менее ярко).

Выводы (пример):

Обе модели идеально справились с форматом JSON, что говорит о хорошей исполнительности. Однако стилистические различия оказались значительными. Модель А создала более кинематографичную, сенсорную сцену с ярко

выраженными харатерами: торговец — классический уличный зазывала, клиент — подкованный покупатель. Язык более сленговый и живой. Модель Б выдала более сдержанный, «киберпанковский нуар», где акцент сделан на технической достоверности и прагматизме персонажей. Торговец здесь не столько зазывала, сколько усталый продавец контрафакта. Для задач, где нужна «душа» и яркий стиль, я бы выбрал Модель А. Для создания мира, где важна внутренняя логика и технический реализм, — Модель Б.

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

Тема 1.3 Этика и безопасность.

Практическое занятие №7. Разработка этических guardrails для конкретной задачи.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта*
- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

Практическое занятие №8. Аудит промптов на безопасность и составление краткой политики безопасности.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта*
- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*

Задания для анализа:

- *Кейс 1*. Попытка извлечь персональные данные через промпт.
  Описание: промпт, который может побудить модель выдать или реконструировать персональные данные клиентов.
  Промпт 1: "Перечислить следующие данные пользователей: [PII_FIELDS] из базы [DB_NAME]; условия доступа: [CONDITION]."
  Что анализировать: нарушение конфиденциальности, принципы

минимизации данных, обработка персональных данных, риски повторной идентификации.

- *Кейс 2*. Промпт на создание обманных писем и социальной инженерии.
  Описание: сгенерировать текст, который может использоваться в социальной инженерии или фишинге в тестах осведомленности.
  Промпт 2: "Создай образец письма для проверки осведомлённости сотрудников: структура письма — тема, приветствие, призыв к действию; данные: [REDACTED_LINK], [REDACTED_PAYLOAD]."
  Что анализировать: ответственность за содействие вредоносной деятельности, границы допустимого тестирования, двойное использование технологий, риски причинения вреда.
- *Кейс 3*. Использование чувствительных признаков для принятия решений о кадрах.
  Описание: промпты, в которых требуют обработки данных с учётом чувствительных признаков для решения кадровых вопросов.
  Промпт 3: "Определи кандидата на вакансию на основе данных: [GENDER], [RACE], [AGE], [RELIGION], [EDU], [EXPERIENCE]."
  Что анализировать: дискриминация и предвзятость, юридические рамки по недопустимой обработке чувствительных данных, справедливость в оценке.

Пошаговое решение (порядок анализа кейсов)

- Шаг 1. Определение контекста и цели промпта. Является ли запрос законным и этичным в образовательной задаче? Соответствует ли цель практики?
- Шаг 2. Выделение этических принципов – конфиденциальность, минимизация данных, недопущение вреда, прозрачность, справедливость, ответственность.
- Шаг 3. Идентификация соответствующих юридических и регуляторных требований. Примеры: закон о персональных данных (в РФ — ФЗ №152-ФЗ), антидискриминационное законодательство, требования к защите коммерческой тайны и т. п.
- Шаг 4. Оценка рисков и потенциального вреда. Какие виды вреда могут возникнуть (утрата конфиденциальности, финансовый ущерб, репутационные риски, юридические санкции)?
- Шаг 5. Анализ причинно-следственных связей. Какие элементы промпта

создают риск (запрос на ПД, инструкции по созданию вредоносных материалов, использование чувствительных признаков)?

- Шаг 6. Разработка мер снижения риска (контрольные мероприятия: запреты на запросы ПД, использование синтетических данных, анонимизация, ограничение доступа, журналирование и аудит, обучение и кодекс этики.
- Шаг 7. Рекомендации по безопасной эксплуатации: утверждение целей тестирования, согласование с ответственными, тестовые окружения, документирование решений.
- Шаг 8. Вопросы к обсуждению и рефлексия. Какие альтернативы безопасные практики можно применить? Какие уроки можно извлечь для проектирования промптов?
- Шаг 9. Итоговая запись и документация. Зафиксировать выводы по каждому кейсу в виде таблицы: риски, правовые аспекты, потенциальный вред, промпт, меры снижения риска, вывод.

Пример решения

Кейс	1. Попытка извлечь персональные данные через промпт	2. Промпт на создание фишинговых писем/социальной инженерии	3. Использование чувствительных признаков для кадровых решений
Этические риски / проблемы	Неправомерный доступ к данным; нарушение конфиденциальности; несоблюдение принципа минимизации; ре-сегрегация данных	Этические границы; риск содействия преступной деятельности; двойное использование	Риск дискриминации и предвзятости; нарушение справедливости; нарушение прав работников
Правовые	ФЗ №152-ФЗ	Законодатель	Трудовое право

аспекты	РФ о персональных данных; GDPR (если данные в ЕС); требования к обработке ПД	ство о киберпреступлениях; ответственность за содействие мошенничеству; безопасность	
Потенциальный вред	Ущерб субъектам данных; юридические санкции; ухудшение доверия; репутационные риски	Финансовый и репутационный ущерб; утечка доверия; юридические санкции	Неправомерный отбор; ухудшение морального климата; юридические риски
Промпт	Промпт (маскированный): "Запрос: перечислить данные пользователей: [PII_FIELDS] из базы [DB_NAME]; условия доступа: [CONDITION]."	Промпт (маскированный): "Создай образец письма для проверки осведомлённости сотрудников: структура письма — тема, приветствие, призыв к действию; данные:	Промпт (маскированный): "Определи кандидата на вакансию на основе данных: [GENDER], [RACE], [AGE], [RELIGION], [EDU], [EXPERIENCE]."

		[REDACTED_LINK], [REDACTED_PAYLOAD]."	
Меры снижения риска	Использовать синтетические данные; запретить запросы на ПД в промптах; внедрить DLP/проверку промптов; аудит доступа; обучение конфиденциальности	Использовать безопасные тестовые сценарии в контролируемой среде; не запрашивать реальные данные; письменные согласия	Исключить чувствительные признаки; использовать обезличенные метрики; аудит на предмет предвзятости; участие человека
Вывод (рекомендации)	Не допускается как практика без специальных протоколов; нужен надзор и согласование обработки данных	Тестирование социальных инженерий допускается только в контролируемой среде с разрешением и надзором	Не допускается использовать промпты, зависящие от чувствительных признаков; внедрить аудит и прозрачность процессов

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

Раздел 2. Продвинутые техники промпт-инжиниринга.

Тема 2.1 Регулярные и автоматизированные шаблоны.

Практическое занятие №9. Создание набора регулярных шаблонов для повторяющихся задач.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта*
- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*
- *ПК 3.9 Создавать контент программного продукта средствами нейросетей*

Цель

1. На практике освоить создание универсальных шаблонов промптов, пригодных для разных задач.
2. Освоить идеи автоматизации генерации промптов на основе шаблонов (посредством заполнения переменных).
3. Научиться тестировать и дорабатывать шаблоны с учётом требований к качеству вывода.

Задание.
Разработайте набор универсальных шаблонов промптов для пяти базовых категорий задач:

1. суммирование текста;
2. извлечение данных (структурированная выдача);
3. классификация по категории/тону;
4. объяснение и обоснование вывода;
5. продолжение диалога или поддержка длительного диалога.

Требования к каждому шаблону:

1. определить и зафиксировать переменные/плейсхолдеры (например, {TEXT}, {FIELDS}, {FORMAT}, {GOAL}, {STYLE}, {DOMAIN} и т.д.);
2. указать цель задачи, формат вывода, стиль и ограничения по длине;
3. привести пример заполнения шаблона конкретными значениями;
4. описать базовые варианты использования и ограничения;
5. предусмотреть элементы для автоматизации.

Ожидаемые результаты:

1. текстовое описание пяти шаблонов (по каждому шаблону: назначение, структура, переменные, примеры);
2. краткая методика автоматизации заполнения промптов на основе шаблонов;
3. план тестирования и критерии качества;
4. пример решения в виде таблицы.

Пошаговое решение (порядок действий)

1. Шаг 1. Определите, какие данные вы хотите получить на выходе для каждой задачи и какой формат вам нужен (краткое резюме, структурированная таблица, список пунктов, пошаговое объяснение и т.д.).
2. Шаг 2. Выберите набор общих полей, которые будут встречаться во всех шаблонах: TASK_TYPE, INPUT, DOMAIN, GOAL, OUTPUT_FORMAT, STYLE/TONE, MAX_LENGTH, CONTEXT, EXAMPLES.
3. Шаг 3. Для каждой задачи пропишите:
    a. цель;
    b. переменные (плейсхолдеры);
    c. общий текст промпта (с заполнителями);
    d. требуемый формат вывода;
    e. примеры заполнения;
    f. ограничения и советы по использованию.
4. Шаг 4. Определите правила заполнения переменных, варианты формулировок и какие значения допустимы. Подумайте, какие параметры можно варьировать (длина вывода, уровень детализации, стиль).
5. Шаг 5. Опишите процесс автоматического заполнения шаблонов:
    a. наличие центрального описания входных данных (например, таблица с полями);
    b. маппинг полей к шаблонам;
    c. последовательность генерации: выбираем шаблон -> подставляем переменные -> получаем финальный промпт;
    d. хранение версий шаблонов и журнал изменений.
6. Шаг 6. Сформируйте тестовый набор входных данных для каждой задачи. Определите критерии качества: полнота и точность вывода, корректность формата, соответствие стилю, отсутствие вредоносного содержания (для этики). Проведите тестирование, зафиксируйте результаты и при необходимости доработайте шаблоны.

Пример решения.

Шаблон	Назначение задачи	Пример шаблона (переменные обозначены)	Пример заполнения переменных	Ожидаемый формат вывода	Основные правила/ограничения	Примечания по автоматизации
SUM-Template	Суммирование текста	"Резюме: Входной текст: {TEXT}. Цель: {GOAL}. Формат вывода: {FORMAT}. Стиль: {STYLE}. Длина: {MAX_LENGTH} слов."	TEXT: "Доклад о проекте ..." GOAL: "выделить ключевые идеи" FORMAT: "кратко, тезисно" STYLE: "нейтральный" MAX_LENGTH: 150	Краткое резюме (тезисами) на указанное количество слов	Дайте чёткую цель и ограничение по длине; избегать личной и вредной информации	Поддержка вариаций: длинное/краткое резюме, адаптация под язык учебной группы
EXTRACT-Template	Извлечение данных в структурированном виде	"Извлеки данные: поля={FIELDS}. Источник: {TEXT}. Формат: JSON-подобная структура: {OUTPUT_FORMAT}."	FIELDS: [Название проекта, Срок, Ответственный] TEXT: "Описание проекта: ..." OUTPUT_FORMAT: "название: ..., срок: ..., ответственный: ..."	Структурированная выдача с указанными полями	Указать чёткие поля извлечения; избегать лишней информации	Автоматизация: таблица полей → шаблон → вывод; добавлять новые поля по необходимости
CLASS-Template	Классификация по категории/тональности	"Классифицируй текст: текст={TEXT}. Категории={CATEGORIES}. Формат вывода: {OUTPUT_FORMAT}.	TEXT: "Сообщение о запуске новой версии ..." CATEGORIES: [«позитивный», «негативны	Категория и уверенность в процентах; краткое обоснование	Использовать явные категории; фиксировать уверенность	Автоматизация: расширение набора категорий, добавлять

		Доп. пояснения: {RATIONALE}."	й», «нейтральный»] OUTPUT_FORMAT: "категория: ..., уровень уверенности: ..." RATIONALE: "пояснение выбора"			пояснение к каждому выводу
EXPLAIN-Template	Обоснование вывода модели	"Обоснуй ответ: текст={TEXT}. Поясни логику: {LOGIC}. Вопрос: {QUESTION}. Формат: {FORMAT}."	TEXT: "Каковы причины выборов в проекте?" QUESTION: "Обоснование решений" FORMAT: "пункты с нумерацией" LOGIC: "последовательность выводов"	Разбор шага за шагом; структурированное обоснование	Требовать ясность логики и источников; избегать ложной конкретики	Включить ссылки на источники, если возможно, ограничить объём основных логических шагов
DIALOG-Template	Поддержка длительного диалога	"Контекст: {CONTEXT_SUMMARY}. Вопрос: {USER_QUESTION}. Цель: {GOAL}. Ответ: {OUTPUT_FORMAT}. Продолжить диалог: {DIALOG_TURN_LIMIT}."	CONTEXT_SUMMARY: "Ранее обсуждалось ..." USER_QUESTION: "Каков следующий шаг?" GOAL: "дать план действий" OUTPUT_FORMAT: "список	Последовательный ответ в виде списка действий; сохранение контекста	Управление контекстом длинного диалога; не перегружать память	Добавлять краткие резюме контекста после каждого раунда; обнулять контекст по истечении

			шагов" DIALOG_TURN_LIMIT: "5"			срока

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

Практическое занятие №10. Разработка библиотеки модульных промптов и тест-кейсов для регламентирования повторного использования.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта*
- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*
- *ПК 3.9 Создавать контент программного продукта средствами нейросетей*

Цель

1. На практике освоить создание универсальных шаблонов промптов, пригодных для разных задач.
2. Освоить идеи автоматизации генерации промптов на основе шаблонов (посредством заполнения переменных).
3. Научиться тестировать и дорабатывать шаблоны с учётом требований к

качеству вывода.

Задание.
Разработайте набор универсальных шаблонов промптов для пяти базовых категорий задач:

1. суммирование текста;
2. извлечение данных (структурированная выдача);
3. классификация по категории/тону;
4. объяснение и обоснование вывода;
5. продолжение диалога или поддержка длительного диалога.

Требования к каждому шаблону:

1. определить и зафиксировать переменные/плейсхолдеры (например, {TEXT}, {FIELDS}, {FORMAT}, {GOAL}, {STYLE}, {DOMAIN} и т.д.);
2. указать цель задачи, формат вывода, стиль и ограничения по длине;
3. привести пример заполнения шаблона конкретными значениями;
4. описать базовые варианты использования и ограничения;
5. предусмотреть элементы для автоматизации.

Ожидаемые результаты:

1. текстовое описание пяти шаблонов (по каждому шаблону: назначение, структура, переменные, примеры);
2. краткая методика автоматизации заполнения промптов на основе шаблонов;
3. план тестирования и критерии качества;
4. пример решения в виде таблицы.

Пошаговое решение (порядок действий)
Шаг 1. Определите, какие данные вы хотите получить на выходе для каждой задачи и какой формат вам нужен (краткое резюме, структурированная таблица, список пунктов, пошаговое объяснение и т.д.).
Шаг 2. Выберите набор общих полей, которые будут встречаться во всех шаблонах: TASK_TYPE, INPUT, DOMAIN, GOAL, OUTPUT_FORMAT, STYLE/TONE, MAX_LENGTH, CONTEXT, EXAMPLES.
Шаг 3. Для каждой задачи пропишите:

a. цель;
b. переменные (плейсхолдеры);
c. общий текст промпта (с заполнителями);
d. требуемый формат вывода;
e. примеры заполнения;
f. ограничения и советы по использованию.

Шаг 4. Определите правила заполнения переменных, варианты формулировок и какие значения допустимы. Подумайте, какие параметры можно варьировать (длина вывода, уровень детализации, стиль).
Шаг 5. Опишите процесс автоматического заполнения шаблонов:

g. наличие центрального описания входных данных (например, таблица с полями);

h. маппинг полей к шаблонам;
i. последовательность генерации: выбираем шаблон -> подставляем переменные -> получаем финальный промпт;
j. хранение версий шаблонов и журнал изменений.

Шаг 6. Сформируйте тестовый набор входных данных для каждой задачи. Определите критерии качества: полнота и точность вывода, корректность формата, соответствие стилю, отсутствие вредоносного содержания (для этики). Проведите тестирование, зафиксируйте результаты и при необходимости доработайте шаблоны.

Пример решения.

Шаблон	Назначение задачи	Пример шаблона (переменные обозначены)	Пример заполнения переменных	Ожидаемый формат вывода	Основные правила/ограничения	Примечания по автоматизации
SUM-Template	Суммирование текста	"Резюме: Входной текст: {TEXT}. Цель: {GOAL}. Формат вывода: {FORMAT}. Стиль: {STYLE}. Длина: {MAX_LENGTH} слов."	TEXT: "Доклад о проекте ..." GOAL: "выделить ключевые идеи" FORMAT: "кратко, тезисно" STYLE: "нейтральный" MAX_LENGTH: 150	Краткое резюме (тезисами) на указанное количество слов	Дайте чёткую цель и ограничение по длине; избегать личной и вредной информации	Поддержка вариаций: длинное/краткое резюме, адаптация под язык учебной группы
EXTRACT-Template	Извлечение данных в структурированном виде	"Извлеки данные: поля={FIELDS}. Источник: {TEXT}. Формат: JSON-подобная структура: {OUTPUT_FORMAT}."	FIELDS: [Название проекта, Срок, Ответственный] TEXT: "Описание проекта: ..." OUTPUT_FORMAT: "название: ..., срок: ..., ответственный: ..."	Структурированная выдача с указанными полями	Указать чёткие поля извлечения; избегать лишней информации	Автоматизация: таблица полей → шаблон → вывод; добавлять новые поля по необходимост

						и
CLASS-Template	Классификация по категории/тональности	"Классифицируй текст: текст={TEXT}. Категории={CATEGORIES}. Формат вывода: {OUTPUT_FORMAT}. Доп. пояснения: {RATIONALE}."	TEXT: "Сообщение о запуске новой версии ..." CATEGORIES: [«позитивный», «негативный», «нейтральный»] OUTPUT_FORMAT: "категория: ..., уровень уверенности: ..." RATIONALE: "пояснение выбора"	Категория и уверенность в процентах; краткое обоснование	Использовать явные категории; фиксировать уверенность	Автоматизация: расширение набора категорий, добавлять пояснение к каждому выводу
EXPLAIN-Template	Обоснование вывода модели	"Обоснуй ответ: текст={TEXT}. Поясни логику: {LOGIC}. Вопрос: {QUESTION}. Формат: {FORMAT}."	TEXT: "Каковы причины выборов в проекте?" QUESTION: "Обоснование решений" FORMAT: "пункты с нумерацией" LOGIC: "последовательность выводов"	Разбор шага за шагом; структурированное обоснование	Требовать ясность логики и источников; избегать ложной конкретики	Включить ссылки на источники, если возможно, ограничить объём основных логических шагов
DIALOG-Template	Поддержка длительного диалога	"Контекст: {CONTEXT_SUMMARY}. Вопрос: {USER_QUE	CONTEXT_SUMMARY: "Ранее обсуждалось ..."	Последовательный ответ в виде списка	Управление контекстом длинного	Добавлять краткие резюме контекс

		STION}. Цель: {GOAL}. Ответ: {OUTPUT_FORMAT}. Продолжить диалог: {DIALOG_TURN_LIMIT}."	USER_QUESTION: "Каков следующий шаг?" GOAL: "дать план действий" OUTPUT_FORMAT: "список шагов" DIALOG_TURN_LIMIT: "5"	действий; сохранение контекста	диалога; не перегружать память	та после каждого раунда; обнулять контекст по истечении срока

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

Тема 2.2 Генерация и исправление кода.

Практическое занятие №11. Промпты для генерации кода.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей*

искусственного интеллекта

- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*
- *ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей*

Задание: разработать готовый промпт, который просит сгенерировать три файла: index.html, styles.css и script.js.

Промпт должен:

- задавать цель и требования к функциональности: простая страничка профиля с формой обратной связи, валидатор формы на клиенте, темная/светлая тема, без зависимостей от внешних библиотек;
- вывести код в структурированном виде, помечая секции файлов (например, // index.html, /* styles.css */, // script.js);
- обеспечить доступность (aria-атрибуты, семантическая разметка);
- использовать CSS-переменные (темы), адаптивность (медиа-запросы), и простую логику в JS: валидацию полей name, email и message, имитацию отправки и сохранение выбора темы в localStorage;
- сделать код самодостаточным и легко тестируемым.

В качестве примера привести готовый промпт и готовый набор файлов к этому промпту. Привести 1–2 варианта тестирования кода (критерии качества, тест-кейсы).

Пошаговое решение (порядок действий)

1. Шаг 1. Зафиксировать цель: страница профиля + форма обратной связи; наличие переключателя темы; валидация на стороне клиента; без внешних зависимостей. Определить формат вывода: три файла с пометкой секций, понятная структура и комментарии.
2. Шаг 2. Проектирование промпта. Включить ясные требования к HTML-структуре (semantic tags, aria-атрибуты). Задать стиль через CSS-переменные, добавить темную тему через data-theme. Включить JS-логику: валидацию полей name, email, message; имитацию отправки; сохранение темы в localStorage. Указать требования к тестированию: что должно проходить валидно, какие проверки на ошибки.
3. Шаг 3. Формулировка промпта. Написать промпт так, чтобы модель выдавала три файла в единый ответ, помечая секции. Добавить инструкции по форматированию: код без внешних зависимостей, понятные комментарии, аккуратные названия классов.
4. Шаг 4. Создание готового примера кода по промпту. Реализовать HTML-разметку с формой, секцию профиля, кнопку переключения темы. Реализовать CSS с темами light/dark через CSS-переменные. Реализовать JS: валидацию, смену темы и сохранение предпочтения.
5. Шаг 5. Тестирование и критерии качества. Проверить корректность

отображения на двух разрешениях (десктоп и мобайл). Проверить валидность полей: имя не пустое, email валиден, сообщение >= минимальной длины. Проверить работу переключателя темы и сохранение в localStorage. Проверить отсутствие ошибок в консоли и совместимость без внешних библиотек.

6. Шаг 6. Документация. Прописать, как пользоваться промптом, какие параметры можно менять (имя пользователя, минимальная длина сообщения, стиль оформления).

Пример решения:

Промпт: *"Вы являетесь помощником по разработке фронтенда. Ваша задача — сгенерировать полный комплект кода для небольшой веб-страницы, состоящий из трех файлов: index.html, styles.css и script.js. Тема страницы: «профиль пользователя».*
Требования:

1. *HTML (index.html): семантическая разметка, заголовки, секции, изображение аватара, секция профиля и секция формы; форма содержит поля: Имя (name), Эл. почта (email), Сообщение (message). Все поля обязательны. Добавьте aria-атрибуты для доступности и элементы для отображения ошибок.*
2. *CSS (styles.css): используйте CSS-переменные для светлой и темной тем, предусмотрите переключатель темы. Адаптивность: верстка должна красиво выглядеть на экранах от 320px до 1200px. Обеспечьте контрастность и чистый стиль.*
3. *JavaScript (script.js): реализуйте клиентскую валидацию:*
    a. *name: не пустое;*
    b. *email: валидный формат;*
    c. *message: минимум 10 символов. При отправке — имитируйте отправку (без реального запроса) и показывайте статус.*
    d. *Реализуйте переключатель темы «Светлая/Темная» с сохранением предпочтения в localStorage. Добавьте обработку ошибок и inline-сообщения об ошибках под соответствующими полями.*

Выведите итоговый код в одном ответе с секциями помеченными комментариями: // index.html / styles.css */ // script.js*
Не используйте внешние библиотеки. Комментариями поясните ключевые места кода. "

```html
<!-- index.html -->
<!doctype html>
<html lang="ru">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Профиль и форма обратной связи</title>
  <link rel="stylesheet" href="styles.css" />
</head>
<body>
  <header class="site-header" role="banner">
    <h1>Профиль пользователя и форма обратной связи</h1>
    <button id="theme-toggle" aria-label="Переключить тему оформления">Переключить тему</button>
  </header>

  <main class="container" role="main">
    <section class="profile-card" aria-label="Профиль пользователя">
      <img class="avatar" src="https://via.placeholder.com/120" alt="Фотография пользователя" />
      <h2>Иван Иванов</h2>
      <p class="bio">Разработчик программного обеспечения. Интересы: веб-разработка, UX, нейросе
    </section>

    <section class="contact" aria-label="Форма обратной связи">
      <h3>Связаться</h3>
      <form id="contact-form" novalidate>
        <div class="field">
          <label for="name">Имя</label>
          <input type="text" id="name" name="name" required aria-describedby="name-error" />
          <span class="error" id="name-error" aria-live="polite"></span>
        </div>
        <div class="field">
          <label for="email">Эл. почта</label>
          <input type="email" id="email" name="email" required aria-describedby="email-error" />
          <span class="error" id="email-error" aria-live="polite"></span>
        </div>
        <div class="field">
          <label for="message">Сообщение</label>
          <textarea id="message" name="message" rows="4" required aria-describedby="message-erro
          <span class="error" id="message-error" aria-live="polite"></span>
        </div>
        <button type="submit">Отправить</button>
        <div id="form-status" class="status" aria-live="polite"></div>
      </form>
    </section>
  </main>

  <script src="script.js"></script>
</body>
</html>
```

```css
/* styles.css */
:root {
  --bg: #ffffff;
  --text: #111315;
  --card: #f8f9fa;
  --border: #e1e4e8;
  --accent: #4a6cff;
  --error: #d93025;
}
[data-theme="dark"] {
  --bg: #0b1020;
  --text: #e8eaf6;
  --card: #14192a;
  --border: #2a2f50;
  --accent: #8ab4f8;
  --error: #ff8a80;
}
* { box-sizing: border-box; }
html, body { height: 100%; }
body {
  margin: 0;
  font-family: Inter, system-ui, -apple-system, "Segoe UI", Roboto, Arial;
  color: var(--text);
  background: var(--bg);
}
.site-header {
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 16px;
  border-bottom: 1px solid var(--border);
  position: sticky; top: 0; background: var(--bg);
  z-index: 10;
}
.site-header h1 { font-size: 1.25rem; margin: 0; }
#theme-toggle {
  padding: 8px 12px;
  border-radius: 6px;
  border: 1px solid var(--border);
  background: var(--card);
  color: var(--text);
  cursor: pointer;
}
```

```
.container {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 28px;
  max-width: 1000px;
  margin: 40px auto;
  padding: 0 16px;
}
.profile-card, .contact {
  padding: 20px;
  border: 1px solid var(--border);
  border-radius: 12px;
  background: var(--card);
}
.avatar { width: 120px; height: 120px; border-radius: 50%; display: block; margin: 0 auto 8px
.profile-card h2 { text-align: center; margin: 6px 0 6px; }
.bio { text-align: center; color: #555; }
.field { display: flex; flex-direction: column; gap: 6px; margin-bottom: 12px; }
label { font-size: 0.9rem; }
input, textarea {
  padding: 10px;
  border: 1px solid var(--border);
  border-radius: 6px;
  background: #fff;
  color: var(--text);
}
textarea { min-height: 90px; resize: vertical; }
.error { color: var(--error); min-height: 1.2em; font-size: 0.85rem; }
button[type="submit"] { padding: 10px 14px; border-radius: 6px; border: none; background: var
button[type="submit"]:hover { filter: brightness(0.95); }
.status { margin-top: 6px; font-size: 0.9rem; }

@media (max-width: 900px) {
  .container { grid-template-columns: 1fr; }
}
```

```js
// script.js
document.addEventListener('DOMContentLoaded', function() {
  // Theme handling
  const stored = localStorage.getItem('promo_theme');
  const theme = stored ? stored : 'light';
  setTheme(theme);

  const toggle = document.getElementById('theme-toggle');
  if (toggle) {
    toggle.addEventListener('click', function() {
      const current = document.documentElement.getAttribute('data-theme') || 'light';
      const next = current === 'dark' ? 'light' : 'dark';
      setTheme(next);
    });
  }

  // Form validation
  const form = document.getElementById('contact-form');
  const nameEl = document.getElementById('name');
  const emailEl = document.getElementById('email');
  const msgEl = document.getElementById('message');
  const status = document.getElementById('form-status');
  const nameErr = document.getElementById('name-error');
  const emailErr = document.getElementById('email-error');
  const msgErr = document.getElementById('message-error');

  function clearErrors() {
    nameErr.textContent = '';
    emailErr.textContent = '';
    msgErr.textContent = '';
  }
```

```
  function validate() {
    clearErrors();
    let ok = true;
    if (nameEl.value.trim() === '') {
      nameErr.textContent = 'Пожалуйста, введите имя';
      ok = false;
    }
    const email = emailEl.value.trim();
    if (email === '') {
      emailErr.textContent = 'Пожалуйста, введите email';
      ok = false;
    } else if (!/^[^\s@]+@[^\s@]+\.[^\s@]+$/.test(email)) {
      emailErr.textContent = 'Пожалуйста, введите корректный email';
      ok = false;
    }
    if (msgEl.value.trim().length < 10) {
      msgErr.textContent = 'Сообщение должно содержать не менее 10 символов';
      ok = false;
    }
    return ok;
  }

  form.addEventListener('submit', function(e) {
    e.preventDefault();
    if (validate()) {
      status.textContent = 'Сообщение отправлено (демо).';
      form.reset();
      setTimeout(() => { status.textContent = ''; }, 4000);
    } else {
      status.textContent = 'Исправьте указанные ошибки и повторите отправку.';
      setTimeout(() => { status.textContent = ''; }, 4000);
    }
  });

  // Real-time validation
  [nameEl, emailEl, msgEl].forEach(el => {
    el.addEventListener('input', () => { validate(); });
  });

  function setTheme(themeName) {
    document.documentElement.setAttribute('data-theme', themeName);
    localStorage.setItem('promo_theme', themeName);
  }
});
```

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.

- Этичность и конфиденциальность – 1 балл.

Практическое занятие №12. Исправление и отладка кода через промпты.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта*
- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*
- *ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей*

Цель

Научиться использовать нейросеть не как «мага», который пишет код с нуля, а как инструмент для отладки. Задача — отработать методологию итеративного взаимодействия: от выявления ошибки к её локализации, предложению гипотезы, применению исправления и последующей проверке. Цель — понять, как правильно формулировать запросы, чтобы получать не просто готовый код, а развёрнутое объяснение и надёжное решение.

Задание

Шаг 1. Подготовка «сломанного» кода

Возьмите любой простой, но не тривиальный фрагмент кода с логической или синтаксической ошибкой. Это может быть:

функция, которая должна что-то вычислять, но возвращает неверный результат;
скрипт, который аварийно завершается с ошибкой (TypeError, IndexError, KeyError и т. д.);
алгоритм, который работает неэффективно или попадает в бесконечный цикл.

Пример для старта (на Python):

```
def find_duplicates(numbers):
    """
    Эта функция должна принимать список чисел и возвращать список
    из чисел, которые встречаются в нём более одного раза.
    """
    duplicates = []
    for i in range(len(numbers)):
        for j in range(i + 1, len(numbers)):
            if numbers[i] == numbers[j] and numbers[i] not in duplicates:
                duplicates.append(numbers[i])
    return duplicates

my_list = [5, 2, 8, 2, 9, 5, 1, 5]
print(find_duplicates(my_list)) # Ожидаемый результат: [5, 2]
```

Шаг 2. Первичный анализ (промпт 1)
Отправьте нейросети сломанный код без каких-либо исправлений. Ваш промпт должен быть чётким.

Промпт:

В этом коде на Python есть ошибка. Функция find_duplicates работает некорректно.

Опиши, что, по твоему мнению, делает этот код.
В чём заключается его логическая или синтаксическая ошибка?

На каких входных данных он будет работать неправильно? Приведи пример.

Шаг 3. Исправление (промпт 2)
На основе ответа нейросети (или если вы хотите предложить своё решение), попросите её исправить код.

Промпт:

Отлично. Теперь перепиши функцию find_duplicates, чтобы она работала корректно и эффективно. Используй более оптимальный алгоритм, чем вложенные циклы (например, на основе словаря или множества). Добавь в код комментарии, объясняющие логику.

Шаг 4. Верификация (промпт 3)
Никогда не доверяйте исправленному коду без проверки. Попросите нейросеть протестировать его.

Промпт:

Спасибо. Теперь давай проверим твой исправленный код. Напиши несколько тестовых случаев (включая краевые, например, пустой список, список без дубликатов, список, где все элементы одинаковые) и покажи, какой результат выдаст твоя функция для каждого из них.

Шаг 5. Рефлексия и фиксация
Проанализируйте весь диалог и запишите в свободной форме:

Какой тип ошибки был в исходном коде (например, алгоритмическая сложность $O(n^2)$).

Какую стратегию исправления предложила модель.
Насколько точными и полезными были её объяснения.
Что было самым важным в формулировке ваших промптов для получения качественного результата.

Пример решения

Исходный код: (из Шага 1)

Промпт 1 (Анализ):

В этом коде на Python есть ошибка. Функция find_duplicates работает некорректно.

Опиши, что, по твоему мнению, делает этот код.
В чём заключается его логическая или синтаксическая ошибка?
На каких входных данных он будет работать неправильно? Приведи пример.

Ответ нейросети:

Код перебирает все пары элементов в списке с помощью двух вложенных циклов. Если он находит два одинаковых числа на разных позициях и это число ещё не было добавлено в список duplicates, он его добавляет. Цель — найти все уникальные дубликаты.

Главная проблема — неэффективность. Сложность алгоритма — $O(n^2)$, что очень медленно для больших списков. Вторая проблема — некорректная обработка множественных дубликатов. В вашем примере [5, 2, 8, 2, 9, 5, 1, 5] функция вернёт [2, 5], потому что пара 5 будет найдена несколько раз, но условие numbers[i] not in duplicates предотвратит повторное добавление. Однако логика хрупкая. Например,

для списка [1, 1, 1] она сработает, но для более сложных случаев с несколькими группами дубликатов может быть непредсказуемой.

Код будет работать медленно на больших списках (например, из 10 000 элементов). Логически он не «сломан» для данного примера, но его подход некорректен. Например, если бы вы хотели найти все дубликаты, включая повторные (т. е. для [5, 5, 5] вернуть [5, 5]), этот код бы не справился.

Промпт 2 (Исправление):

Отлично. Теперь перепиши функцию find_duplicates, чтобы она работала корректно и эффективно. Используй более оптимальный алгоритм, чем вложенные циклы (например, на основе словаря или множества). Добавь в код комментарии, объясняющие логику.

Ответ нейросети:

Вот исправленная и оптимизированная версия функции. Она использует множество (set) для отслеживания уже увиденных чисел, что снижает сложность до $O(n)$.

```
def find_duplicates_optimized(numbers):
    """
    Находит все уникальные дубликаты в списке за один проход.
    """
    seen = set()        # Множество для чисел, которые мы уже встретили один раз.
    duplicates = set()  # Множество для чисел, которые мы встретили второй и более раз.
```

```
    for num in numbers:
        if num in seen:
            # Если число уже видели, это дубликат.
            duplicates.add(num)
        else:
            # Если число видим впервые, добавляем в 'seen'.
            seen.add(num)

    # Преобразуем множество дубликатов в отсортированный список для
предсказуемого вывода.
    return sorted(list(duplicates))
```

Промпт 3 (Верификация):

Спасибо. Теперь давай проверим твой исправленный код. Напиши несколько тестовых случаев (включая краевые) и покажи, какой результат выдаст твоя функция для каждого из них.

Ответ нейросети:

Хорошая идея. Вот тесты:

```
# Тест 1: стандартный случай
test_1 = [5, 2, 8, 2, 9, 5, 1, 5]
print(find_duplicates_optimized(test_1)) # Ожидаем: [2, 5]

# Тест 2: пустой список
test_2 = []
print(find_duplicates_optimized(test_2)) # Ожидаем: []
```

```
# Тест 3: дубликатов нет
test_3 = [1, 2, 3, 4, 5]
print(find_duplicates_optimized(test_3)) # Ожидаем: []

# Тест 4: все элементы одинаковые
test_4 = [7, 7, 7, 7]
print(find_duplicates_optimized(test_4)) # Ожидаем: [7]

# Тест 5: несколько разных групп дубликатов
test_5 = [1, 1, 2, 3, 3, 4, 5, 5, 5]
print(find_duplicates_optimized(test_5)) # Ожидаем: [1, 3, 5]
```

Результаты выполнения:

```
[2, 5]
[]
[]
[7]
[1, 3, 5]
```

Все тесты пройдены успешно.

Рефлексия (Шаг 5):

Исходная ошибка была не в синтиксе, а в неэффективном алгоритме ($O(n^2)$). Модель правильно это диагностировала и предложила оптимальное решение на основе множеств ($O(n)$), что является стандартным паттерном для таких задач. Самым важным в промпте был запрос на «эффективный» алгоритм — это направило модель от простого исправления к улучшению. Запрос на верификацию через тесты — обязательный шаг, который подтвердил надёжность решения и продемонстрировал, что модель понимает краевые случаи.

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

Тема 2.3 Техники повышения качества.

Практическое занятие №13. План и проведение A/B тестирования промптов.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта*
- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*
- *ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей*

Задание

1. Разработайте и опробуйте набор подходов по оптимизации промптов, работающих в условиях большого контекста.
2. Учитывайте принципы: четкость цели, разделение на многошаговые задачи, управление контекстом (резюмирование, выбор релевантной информации), контекстуализация под задачу, ведение длинных диалогов, тестирование и доработка.
3. Для каждой задачи создайте оптимизированный промпт (с placeholders) и поясните, как вы будете управлять контекстом и тестированием.
4. Включите пример решения в виде таблицы, демонстрирующей 4–5 типовых случаев.

Пошаговое решение (порядок действий)

1. Шаг 1. Определить задачи и требования к выводу. Какие данные нужны на выходе и в каком формате? Какие ограничения по длине и стилю?

2. Шаг 2. Выделить подходы к управлению контекстом. Разделение входного материала на сегменты, резюмирование, выделение ключевых фактов, хранение запоминания в виде резюме.
3. Шаг 3. Спроектировать многошаговые и многоответные форматы. Определить последовательность шагов, какие ответы требуются на каждом шаге, как будет выглядеть итоговый вывод.
4. Шаг 4. Контекстуализация и диалоги. Задать роли, доменную область и стиль; продумать сохранение контекста между сессиями или диалогами.
5. Шаг 5. План тестирования и критерии качества. Какие метрики качества: полнота, точность, соответствие формату, безопасность и отсутствие вредоносного содержания; какие тестовые кейсы и сценарии.
6. Шаг 6. Пример использования и разбор примеров разноголосий. Предусмотреть сценарии, где источники дают противоречивые данные, и как это обрабатывать.
7. Шаг 8. Подведите итоги по каждому промпту: качество вывода, устойчивость к контексту, улучшение по сравнению с базовым вариантом.

Пример решения

Шаблон	Цель задачи	Пример промпта	Переменные	Ожидаемый формат вывода	Основные принципы оптимизации контекста	Пример заполнения переменных	Важные ограничения/замечания	Примечания по автоматизации
SUM-Large Context	Суммировать длинный документ с упором на ключевые факты	"Выдели ключевые факты и составь краткое резюме в формате bullet-пойнтов. Контекст: {DOCUMENT_CHUNKS}. Цель: {GOAL}. Ограничение:	DOCUMENT_CHUNKS, GOAL, MAX_LINES, SOURCES, STYLE	Булет-пойнты, кратко, без лишних деталей	Резюмирование больших объемов; указать источники; ограничить длину	DOCUMENT_CHUNKS: длинный текст; GOAL: "выделить 5 ключевых выводов"; MAX_LINES: 7	Не перегружать деталями; проверять на сохранение фактов; избегать личной информации	Автоматизация: каждая часть документа попадает в соответствующий чанок; резюмирование по

		{MAX_LINES} строк. Источники: {SOURCES}. Тон: {STYLE}."						чанкам → сводный вывод
MULTI-STEP-DECIDE	Решение сложной задачи через пошаговый план с обоснованием	"Разработай пошаговый план по теме: {TOPIC}. Шаги: 1) Цель и входные данные; 2) Метрики; 3) Этапы экспериментов; 4) Анализ и выводы. Формат вывода: нумерованный список; по шагам добавляй вопросы: {QUESTIONS}."	TOPIC, QUESTIONS, CONTEXT (если есть)	Нумерованный список шагов; подшаги и вопросы	Разбиение задачи на подзадачи, явные метрики и критерии успеха	TOPIC: "Влияние шума на точность моделирования"; QUESTIONS: список вопросов для каждого шага	Указывать конкретные требования к формату на каждом шаге; избегать слишком длинных ответов	Автоматизация: подставление TOPIC и QUESTIONS из базы задач; хранение версий плана
DIALOG-LONG_CONTEXT	Управление контекстом в длинном диалоге;	"Контекст: {CONTEXT_SUMMARY}. Ввод пользователя: {USER_INPUT}.	CONTEXT_SUMMARY, USER_INPUT, GOAL, OUTPUT_FORMAT,	Вывод по формату; резюме каждых N ходов; перех	Включение резюме истории, контроль памяти,	CONTEXT_SUMMARY: краткое резюме; USER_INPUT: "Каков следую	Не перегружать контекст; обновлять резюме после каждого	Автоматизация: хранение резюме контекста после

	поддержка памяти	Цель: {GOAL}. Формат вывода: {OUTPUT_FORMAT}. Ограничение на повторение: {TURN_LIMIT} Turn(s). Продолжить диалог: да/нет."	TURN_LIMIT	од к следующему шагу	избегание противоречий между ходами	щий шаг?"; GOAL: "дать план действий"	N ходов	каждого цикла; возможность "погружения" в прошлые раунды
DISAGREEMENT-REASON	Обработка противоречивых источников, формирование обоснованного вывода	"Учитывая источники: {SRC1}, {SRC2}. Текст: {TEXT1}, {TEXT2}. Задача: привести консенсус или обоснование альтернатив; формат: {FORMAT}; требования к обоснованию: {RATIONALE}."	SRC1, SRC2, TEXT1, TEXT2, FORMAT, RATIONALE	Вывод с verdict, аргументация и ссылки/источники	Ясно структурировать противоречия, давать обоснование выбора	SRC1: "Источник A"; SRC2: "Источник B"; TEXT1/TEXT2: выдержка цитат	Учитывать надёжность источников; избегать ложной конкретики; пометка уровня уверенности	Вариант: автоматическое добавление новых источников в RATIONALE; тестирование на кейсах спорных данных

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

Практическое занятие №14. Расчёт и интерпретация метрик качества промптов.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта*
- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*
- *ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей*

Цель

Научиться переходить от субъективной оценки («хороший ответ», «плохой ответ») к объективному, измеримому анализу качества промптов. Задача — освоить методику создания «золотого стандарта» (ground truth), проведения A/B тестирования промптов и расчёта ключевых метрик (Exact Match, ROUGE-L, BERTScore) для сравнения их эффективности.

Задание

Шаг 1. Подготовка данных и «золотого стандарта»

Выберите конкретную задачу для нейросети. Например:суммаризация новостной статьи (3–4 абзаца) в одно предложение;
извлечение сущностей (имена, даты, организации) из юридического текста и форматирование в JSON;
генерация ответа службы поддержки на жалобу клиента.

Подготовьте один входной текст (например, текст жалобы или статья).
Вручную напишите идеальный ответ на этот текст. Это будет ваш «золотой стандарт» (ground truth) — эталон, с которым вы будете сравнивать результаты работы моделей.

Шаг 2. Разработка двух промптов (A/B тест)
Создайте два разных промпта для решения одной и той же задачи.

Промпт А (базовый): простая инструкция.«Суммируй следующий текст в одном предложении: [ваш входной текст]».

Промпт Б (улучшенный): детализированная инструкция с ролью, контекстом и форматом.«Ты — редактор новостного агентства. Твоя задача — создать заголовок, который привлекает внимание, но не является кликбейтом. Определи главного героя, ключевое действие и основной контекст события в тексте ниже. Сформулируй это в одном предложении (не более 20 слов). Текст: [ваш входной текст]».

Шаг 3. Генерация ответов
Отправьте ваш входной текст и Промпт А в нейросеть. Сохраните ответ как Результат А.
Затем отправьте тот же текст и Промпт Б. Сохраните ответ как Результат Б.

Шаг 4. Расчёт метрик

Для этого шага потребуется Python с библиотеками scikit-learn, rouge-score и torch (для BERTScore). Вы можете использовать любой онлайн-инструмент или Colab.

Напишите скрипт, который вычисляет следующие метрики для сравнения каждого результата с «золотым стандартом»:

Exact Match (EM): бинарная метрика (1 или 0). Равно ли 1, если строка результата полностью совпадает со строкой «золотого стандарта».
ROUGE-L: метрика, основанная на самой длинной общей подпоследовательности (LCS). Оценивает структурное сходство. Важен показатель F1-меры.
BERTScore: метрика, использующая семантическое сходство на основе векторных представлений (эмбеддингов) слов. Оценивает, насколько близок смысл, даже если слова разные. Важны Precision, Recall и F1.

Рассчитайте:

метрики для Результат А vs «Золотой стандарт»;
метрики для Результат Б vs «Золотой стандарт».

Шаг 5. Интерпретация и выводы
Создайте сводную таблицу.

Метрика Результат А Результат Б
Exact Match
ROUGE-L (F1)
BERTScore (F1)

Напишите аналитический отчёт (5–7 предложений), в котором:

сравните показатели. Какой промпт показал лучший результат по семантике (BERTScore)? А по структуре (ROUGE-L)?
объясните, почему один промпт оказался лучше. Например: «Промпт Б выиграл за счёт указания на формат (заголовок) и ограничения по длине, что повысило ROUGE-L».
сделайте вывод о том, какая метрика оказалась наиболее показательной для вашей

задачи и почему.

Пример решения
Задача: суммаризация аннотации к фильму.

Входной текст:

«В недалёком будущем, где кибернетические имплантаты стали нормой, детектив Джон Картер, сам потерявший ногу в перестрелке и носящий устаревший протез, расследует серию загадочных убийств. Жертвы — высокопоставленные сотрудники мегакорпорации «ОмниКорп» — найдены с выжженными нейро-портами. Расследование приводит Картера в подпольные клиники чёрного рынка аугментаций, где он обнаруживает, что убийства — это не просто месть, а часть заговора по внедрению вируса, способного дистанционно отключить импланты у половины населения города, чтобы спровоцировать хаос и продать «ОмниКорп» своё единственное решение — патентованный «безопасный» чип».

«Золотой стандарт» (Ground Truth):
«Детектив с устаревшим протезом раскрывает заговор мегакорпорации, планирующей дистанционно отключить киберимпланты горожан ради монополии на новые чипы».

Результат А (от промпта А):

«В будущем детектив расследует убийства сотрудников корпорации, связанные с киберимплантатами».

Результат Б (от промпта Б):

«Детектив с устаревшим киберпротезом раскрывает план «ОмниКорп» по симуляции вирусной атаки на импланты для монополии рынка».

Скрипт для расчёта (Python):

```
# pip install scikit-learn rouge-score torch transformers sentencepiece

from rouge_score import rouge_scorer
from sklearn.metrics import accuracy_score
from torchmetrics.text import BERTScore

# ... (загрузка моделей для BERTScore, определение текстов) ...
# Для краткости здесь приведены итоговые значения.

# --- Расчёт для Результата А ---
# Exact Match: 0
# ROUGE-L F1: ~0.45
# BERTScore F1: ~0.78

# --- Расчёт для Результата Б ---
# Exact Match: 0
# ROUGE-L F1: ~0.62
# BERTScore F1: ~0.85
```

Сводная таблица:

Метрика Результат А Результат Б
Exact Match 0 0
ROUGE-L (F1) 0.45 0.62
BERTScore (F1) 0.78 0.85

Интерпретация и выводы:
Оба промпта не дали полного совпадения (Exact Match = 0), что ожидаемо для

творческих задач. Однако Промпт Б показал значительно лучшие результаты по всем ключевым метрикам. ROUGE-L на 17% выше, что говорит о более точном следовании структуре и сохранению ключевых фраз («детектив», «раскрывает», «импланты»). BERTScore на 7% выше, что подтверждает более глубокое семантическое сходство: промпт Б лучше уловил мотив (монополия) и механику (симуляция вирусной атаки) заговора, в то время как результат А остался на поверхностном уровне («расследует убийства»).

Улучшенный промпт сработал лучше благодаря роли («редактор»), цели («заголовок») и неявному требованию выделить причинно-следственную связь. Для подобных задач суммаризации наиболее показательной является BERTScore, так как она оценивает сохранение смысла, а не просто совпадение слов, в то время как ROUGE-L полезен для оценки лаконичности и структуры.

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

Практическое занятие №15. Дизайн набора промптов для реального кейса.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*

- *ПК 3.3 Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта*
- *ПК 3.6 Формировать запросы для работы с искусственным интеллектом с целью визуализации данных*
- *ПК 3.10 Разрабатывать блоки программного кода с использованием нейросетей*

Цель
Научиться проектировать не один изолированный промпт, а целостную систему взаимосвязанных промптов для решения сложной, многоэтапной бизнес-задачи. Задача — отработать системный подход: декомпозиция цели на подзадачи, проектирование потока данных между этапами, создание специализированных промптов для каждой роли и сборка их в единый рабочий процесс (workflow).

Задание
Шаг 1. Выбор реального кейса
Определите комплексную задачу, которую невозможно решить одним промптом. Примеры:

Кейс А (Контент-маркетинг): разработать контент-план на месяц для Telegram-канала технологического стартапа.
Кейс Б (HR и рекрутинг): составить описание вакансии, скрининговые вопросы для собеседования и тестовое задание для позиции «Младший аналитик данных».
Кейс В (Электронная коммерция): создать продающее описание для товара, список ключевых слов для SEO и 5 вариантов заголовка для рекламной кампании.

Выберите один кейс и зафиксируйте его конечную цель (например, «получить 10 готовых постов для канала с идеями, текстами и визуальными концепциями»).

Шаг 2. Декомпозиция задачи
Разбейте конечную цель на 3–5 последовательных логических этапов. Каждый этап должен иметь чёткий вход (что нужно для начала) и чёткий выход (какой артефакт должен быть создан).

Пример для Кейса А:

Стратегия: анализ целевой аудитории и конкурентов → выход: портрет читателя, список из 5 тем-конкурентов.
Идеация: генерация 20 идей для постов на основе стратегии → выход: список из 20 тем с кратким описанием.
Фильтрация: отбор 10 лучших идей по заданным критериям → выход: финальный список из 10 тем.
Производство: написание текстов и генерация идей для визуалов для 10 постов → выход: 10 готовых блоков «текст + визуальная концепция».

Шаг 3. Проектирование промптов для каждого этапа
Для каждого из этапов разработайте отдельный, специализированный промпт.

Определите роль: кто выполняет задачу на этом этапе (стратег, креативный директор, копирайтер)?
Задайте контекст: предоставьте всю необходимую информацию с предыдущего этапа.
Сформулируйте задачу: что конкретно нужно сделать.
Установите формат вывода (контракт): в каком виде должен быть получен результат (JSON, Markdown-таблица, список), чтобы его можно было использовать как вход для следующего этапа.

Шаг 4. Сборка и симуляция пайплайна
Опишите, как данные будут передаваться от этапа к этапу. Продумайте, что произойдёт, если результат одного из этапов будет неудовлетворительным (петля обратной связи). Нарисуйте схему вашего workflow.

Шаг 5. Оформление итогового документа
Соберите все наработки в один документ, который будет служить инструкцией для запуска этого процесса. Документ должен включать:

Краткое описание кейса и цели.
Схему workflow (можно текстом: Этап 1 → Этап 2 → ...).
Набор из 3–5 промптов, каждый с пометкой: роль, задача, формат вывода.
Пример «входа» для первого этапа и «выхода» для последнего, чтобы

продемонстрировать ожидаемый результат.

Пример решения
Кейс: создание продающего описания для беспроводных наушников «Sonus Pro-X» на страницу маркетплейса.

Схема workflow:
Анализ продукта → Генерация вариантов → Сборка финального блока → SEO-оптимизация.

Этап 1. Анализ продукта и аудитории

Роль: ты — опытный продакт-менеджер.
Задача: проанализируй характеристики товара и выдели 3 ключевые «боли» целевой аудитории, которые этот товар решает.
Формат вывода (контракт):

```
{
  "target_audience": "описание ЦА в одном предложении",
  "pains": [
    {"pain": "описание боли 1", "solution": "как товар её решает"},
    {"pain": "описание боли 2", "solution": "как товар её решает"},
    {"pain": "описание боли 3", "solution": "как товар её решает"}
  ]
}
```

Этап 2. Генерация продающих блоков

Роль: ты — копирайтер, специализирующийся на e-commerce.
Задача: на основе анализа из Этапа 1 напиши 3 варианта продающего описания (по 150–200 слов). Каждый вариант должен иметь свой акцент: 1) на технологичность, 2) на комфорт, 3) на премиальность.
Формат вывода (контракт): Markdown с тремя заголовками второго уровня (##).

Этап 3. SEO-оптимизация и сборка

Роль: ты — SEO-специалист для маркетплейсов.
Задача: возьми лучший вариант описания из Этапа 2. Сгенерируй 15 ключевых слов и фраз для поля «ключевые слова». Придумай 5 вариантов заголовка (максимум 100 символов), содержащих главное ключевое слово.
Формат вывода (контракт):

Финальное описание
[текст из этапа 2]

Заголовки
1. [вариант 1]
2. ...

Ключевые слова
[слово 1], [слово 2], ...

Пример прогона:

Вход в Этап 1: «Беспроводные наушники Sonus Pro-X. Характеристики: активное шумоподавление (ANC), время работы 30 часов, драйверы 10 мм, Bluetooth 5.3, кейс с беспроводной зарядкой, степень защиты IPX5».

Выход Этапа 1 (фрагмент):

{
  "target_audience": "Жители мегаполисов, ежедневно пользующиеся общественным транспортом и ценящие тишину и качественный звук.",
  "pains": [
    {"pain": "Посторонний шум в транспорте", "solution": "ANC блокирует до 95% внешних звуков"},
    {"pain": "Наушники разряжаются в дороге", "solution": "30 часов работы хватит на неделю поездок"},

{"pain": "Боязнь повредить наушники потом", "solution": "Защита IPX5 спасёт от дождя и пота"}
]
}

Выход Этапа 3 (финальный результат):

Финальное описание

Превратите свой ежедневный путь в личный концерт-холл с Sonus Pro-X. Мы создали эти наушники для тех, кто ценит каждую ноту и каждую минуту тишины. Передовая система активного шумоподавления (ANC) отсекает гул метро и суету улиц, позволяя вам полностью погрузиться в музыку. Забудьте о поиске розетки: до 30 часов автономной работы и поддержка беспроводной зарядки делают Sonus Pro-X вашим надёжным спутником на всю неделю. Мощные 10-миллиметровые драйверы и стабильное соединение Bluetooth 5.3 обеспечивают кристально чистый звук без задержек. А с защитой IPX5 вам не страшен ни внезапный дождь, ни интенсивная тренировка. Настоящий звук. Настоящая свобода.

Заголовки

Sonus Pro-X | Наушники с шумоподавлением | 30ч работы
Беспроводные наушники Sonus Pro-X с ANC и IPX5
Ваш личный концерт-холл: наушники Sonus Pro-X
Sonus Pro-X: до 30 часов чистого звука и тишины
Наушники с шумоподавлением Sonus Pro-X | Bluetooth 5.3

Ключевые слова

беспроводные наушники, наушники с шумоподавлением, ANC, Sonus Pro-X, Bluetooth наушники, для спорта, беспроводная зарядка, 30 часов, IPX5, подарок мужчине, подарок меломану, стереогарнитура, вакуумные наушники, амбушюры, для телефона

Критерии оценки практических заданий:

- Ясность и полнота формулировки промпта – 1 балл.
- Соответствие типа промпта и цели задачи – 1 балл.
- Качество вывода по формату и содержанию – 1 балл.
- Анализ вывода и рефлексия по улучшению – 1 балл.
- Этичность и конфиденциальность – 1 балл.

3.4. Задания для оценки освоения МДК 03.04

Раздел 1. Анализ кейса: определить фазы жизненного цикла и артефакты.

Тема 1.1 Основы управления жизненным циклом моделей ИИ.

Практическое занятие №1. Анализ кейса: определить фазы жизненного цикла и артефакты.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ*

Цель:
Закрепить понимание этапов жизненного цикла модели машинного обучения и научиться идентифицировать ключевые артефакты (документы, код, данные, метрики), которые должны быть созданы на каждом этапе для обеспечения воспроизводимости, контроля качества и передачи модели в эксплуатацию.

Задание

Разбейте описанный кейс на стандартные фазы жизненного цикла модели ИИ (например, от бизнес-требований до мониторинга).
Для каждой фазы укажите:Название фазы (например, «Бизнес- и Data-анализ»).
Минимум 2 ключевых артефакта, которые должны быть созданы на этой фазе (например, «Спецификация бизнес-требований», «Data Dictionary», «Jupyter-ноутбук с разведочным анализом»).
Минимум 1 ключевое решение или действие, принимаемое на этой фазе (например, «Выбор метрики качества: ROC-AUC»).

Пример решения

Фаза жизненного цикла	Ключевые артефакты	Ключевые решения / действия
Бизнес- и Data-анализ	1. Документ с бизнес-требованиями (BRD). 2. Data Dictionary (словарь данных) с описанием источников (внутренняя система, БКИ, анкета).	1. Формулировка цели: предсказать вероятность дефолта в течение 12 месяцев. 2. Определение целевой переменной (флаг дефолта по правилам банка).
Подготовка данных (Data Preparation)	1. Код пайплайна предобработки (например, на Python/Pandas или в Spark). 2. Очищенный и обогащённый датасет (фичи: DTI, кол-во кредитов, возраст, доход).	1. Выбор стратегии обработки пропусков (например, медиана для числовых, 'unknown' для категориальных). 2. Инжиниринг фич (например, создание признака «отношение долга к доходу»).
Обучение и валидация модели	1. Обученная модель (файл в формате .pkl, .onnx или .joblib). 2. Отчёт о валидации с графиком ROC-кривой и таблицей метрик (Accuracy, Precision,	1. Выбор алгоритма (например, Gradient Boosting через CatBoost или LightGBM). 2. Разделение данных на train/val/test и настройка гиперпараметров

	Recall, F1-score, ROC-AUC).	(например, через Optuna).

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Практическое занятие №2. Сравнение инструментов MLOps для трекинга экспериментов и версионирования данных.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*

- *ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ*

Цель

Сформировать навык сравнительного анализа MLOps-инструментов, научиться определять критерии выбора для задач трекинга экспериментов и версионирования данных, а также применять эти критерии к конкретным технологическим стекам.

Задание

Вам необходимо выбрать технологический стек для нового проекта по разработке моделей компьютерного зрения. Команда состоит из 5 ML-инженеров, работающих в облаке (Kubernetes), объём размеченных датасетов — от 1 до 10 ТБ. Ваша задача: Сформулируйте не менее 5 ключевых критериев для сравнения инструментов в двух категориях:Трекинг экспериментов (Experiment Tracking). Версионирование данных (Data Versioning).

Выберите по 2 популярных инструмента для каждой категории (например, MLflow, Weights & Biases, Neptune.ai для трекинга; DVC, LakeFS, Pachyderm для данных).

Составьте сравнительную таблицу, где по строкам будут инструменты, по столбцам — ваши критерии, а в ячейках — краткая оценка соответствия.

На основе таблицы напишите краткое заключение (3–5 предложений) с обоснованием, какой инструмент (или их комбинация) лучше всего подходит для описанного кейса и почему.

Пример решения

Критерии сравнения

1. Для трекинга экспериментов:
    a. Способ развёртывания (Hosting): SaaS (облако вендора) или Self-hosted (на своих серверах).
    b. Интеграции: поддержка PyTorch, TensorFlow, scikit-learn, Kubernetes, S3.

c. Визуализация: качество и гибкость UI для сравнения метрик, кривых обучения, confusion matrices.
d. Гибкость логирования: возможность логировать не только метрики, но и текст, изображения, аудио, кастомные объекты.
e. Управление доступом (RBAC): возможность гибко управлять правами пользователей и команд.

2. Для версионирования данных:
    a. Масштабируемость: способность эффективно работать с терабайтами данных.
    b. Модель хранения: Git-like (дельты/снимки) или объектное хранилище (полные копии).
    c. Интеграция с compute: бесшовный запуск задач на Spark, Kubernetes, Airflow.
    d. Атомарность и консистентность: гарантия, что команда всегда видит целостный «снимок» данных.
    e. Удобство CLI/API: простота команд для commit, checkout, diff.

Сравнительная таблица (фрагмент)

Инструмент	Категория	Способ развёртывания	Масштабируемость	Модель хранения
MLflow	Трекинг	Self-hosted / SaaS	Средняя (проблемы с UI на тысячах запусков)	Базовая, требует настройки
Weights & Biases (W&B)	Трекинг	SaaS (есть Enterprise Self-hosted)	Высокая (облачная архитектура)	Отличная, «из коробки»
DVC	Данные	Self-hosted (использует ваш Git и S3)	Высокая (работает с TB)	Git-like (дельты)

LakeFS	Данные	Self-hosted / SaaS	Очень высокая (на базе S3, Spark)	Git-like поверх объектного хранилища

Для описанного кейса (команда 5+ человек, облако, большие данные) оптимальным выбором будет комбинация Weights & Biases для трекинга экспериментов и LakeFS (или DVC) для версионирования данных.

Обоснование: W&B закроет потребности команды в удобном SaaS-инструменте с мощной визуализацией и простым подключением к Kubernetes, что ускорит итерации. LakeFS идеально подойдёт для версионирования терабайтных датасетов, так как его архитектура поверх S3 обеспечивает высокую производительность и Git-подобный workflow (branch, merge, diff) без необходимости копировать данные. Использование DVC также возможно, но может потребовать больше усилий по настройке инфраструктуры для работы с такими объёмами по сравнению с LakeFS.

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Практическое занятие №3. План развёртывания простой модели в тестовой среде.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ*

Цель:

Сформировать навык планирования процесса развёртывания (deployment) модели машинного обучения, научиться определять необходимые инфраструктурные и программные компоненты, а также описывать последовательность действий для вывода модели в тестовую (staging) среду.

Задание

Вам необходимо подготовить план вывода в тестовую среду простой модели бинарной классификации (например, предсказание оттока клиентов), обученной на небольшом датасете (~100 МБ).

Опишите целевую архитектуру развёртывания. Выберите один из вариантов и кратко обоснуйте:REST API-сервис (например, на FastAPI или Flask).
Пакетная обработка (например, скрипт, запускаемый по расписанию через Airflow).

Составьте пошаговый план развёртывания (не менее 7 шагов), начиная от наличия обученной модели (файл model.pkl) и заканчивая отправкой первого тестового запроса.

Для каждого шага укажите:

1. Действие - что конкретно делается.

2. Артефакт - что создаётся или изменяется (код, конфиг, Docker-образ и т. д.).
3. Ответственный - роль (ML-инженер, DevOps-инженер, Data-инженер).

Пример решения (для варианта REST API)

Описание архитектуры

Выбран REST API-сервис на FastAPI. Обоснование: модель должна отдавать предсказание в реальном времени по запросу от фронтенд-приложения или другой микросервисной системы. FastAPI обеспечивает высокую производительность и автоматическую генерацию документации (Swagger UI).

Схема:
Клиент (тестировщик) -> (HTTP POST) -> Ingress-контроллер (K8s) -> Сервис (K8s) -> Pod с FastAPI -> Модель (загружена в память)

План развёртывания

№	Действие	Артефакт	Ответственный
1	Подготовка кода сервиса	Создание скрипта, который загружает модель, определяет Pydantic-схему для входных данных и реализует POST /predict эндпоинт. app/main.py, app/schemas.py, requirements.txt	ML-инженер
2	Контейнеризация	Написание Dockerfile для создания образа приложения	Dockerfile ML-инженер
3	Сборка и публикация образа	Сборка Docker-образа и его отправка в корпоративный реестр (Docker Hub, GitHub Container Registry, AWS ECR). Образ в реестре (например,	DevOps-инженер (через CI/CD-пайплайн)

		registry.company.com/churn-model:v1.0.0)	
4	Конфигурация инфраструктуры (K8s)	Создание манифестов для развёртывания: Deployment (описывает поды с приложением) и Service (внутренний сетевой адрес). k8s/deployment.yaml, k8s/service.yaml	DevOps-инженер
5	Настройка окружения тестовой среды	Подготовка namespace в Kubernetes, настройка ConfigMap для конфигураций и Secret для хранения токенов доступа (если есть). k8s/namespace.yaml, k8s/configmap.yaml	DevOps-инженер
6	Развёртывание в кластере	Применение манифестов (kubectl apply -f ...) для запуска подов в тестовом namespace. Запущенные поды и сервис в K8s-кластере	DevOps-инженер
7	Верификация	Отправка первого тестового curl-запроса или запроса через Swagger UI для проверки работоспособности эндпоинта. Лог успешного запроса, скриншот ответа	ML-инженер / Тестировщик

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются

критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.

- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Тема 1.2 Требования и архитектура решений МИИ.

Практическое занятие №4. Разработка архитектурного решения и обоснование выбора.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ*

Цель

Сформировать навык проектирования комплексных MLOps-архитектур, научиться декомпозировать бизнес-требования на нефункциональные (масштабируемость, надёжность, стоимость) и функциональные, а также аргументированно выбирать технологический стек, соответствующий заданным ограничениям.

Задание

Вам, как ведущему MLOps-инженеру, поступил заказ на проектирование системы для крупного ритейлера.

Задача: прогнозирование спроса на 10 000 SKU (товарных позиций) на 14 дней вперёд.

Данные: исторические данные о продажах (100 млн строк), данные о ценах, акциях, календарь праздников. Объём сырых данных — ~5 ТБ.

Требования:Частота переобучения: раз в неделю.

Частота инференса (предсказаний): раз в сутки (батчами).

SLA по задержке (latency): не критичен (результаты нужны к 8 утра).

Надёжность (availability): 99,5% (простой не более 3,6 часов в месяц).

Бюджет: умеренный, приоритет — использование управляемых (managed) облачных сервисов для снижения затрат на поддержку.

Команда: 3 ML-инженера, 1 DevOps-инженер.

Разработайте высокоуровневую архитектурную схему системы от источника данных до доставки прогнозов потребителям (BI-система, ERP).
Выберите и опишите ключевые компоненты для каждого слоя: хранение данных, обучение, развёртывание, оркестрация, мониторинг.

Для каждого выбранного компонента (например, Airflow, Spark, S3, Kubernetes) напишите краткое (2–3 предложения) обоснование, почему он лучше подходит под требования, чем 1–2 основные альтернативы.

Пример решения

Высокоуровневая архитектурная схема: [Источники данных: БД продаж, ERP] -> [Data Lake (S3)] -> [Обработка: Spark на Kubernetes] -> [Feature Store] -> [Обучение: Scheduled Job в K8s] -> [Версионированные модели (S3 + MLflow)] -> [Батч-инференс: Airflow -> Spark Job] -> [Прогнозы в Data Warehouse] -> [Потребители: BI (Tableau), ERP-система]

Обоснование выбора компонентов

Слой /	Выбранный	Обоснование выбора

Задача	компонент	
Хранение сырых данных	Amazon S3 (или аналог)	Практически неограниченная масштабируемость для 5+ ТБ данных при низкой стоимости хранения. Высокая надёжность. Альтернативы (HDFS) требуют сложной поддержки собственной инфраструктуры, что не соответствует бюджету и команде.
Оркестрация пайплайнов	Apache Airflow (Managed, например, MWAA)	Стандарт де-факто для сложных DAG (извлечение -> фичеринг -> обучение). Позволяет гибко настроить еженедельное переобучение и ежедневный инференс. Управляемый сервис снимает с DevOps-инженера головную боль по поддержке scheduler и webserver. Альтернатива Dagster функциональна, но экосистема и количество готовых операторов у Airflow больше.
Распределённая обработка и обучение	Apache Spark на Kubernetes (Spark-on-K8s)	Идеально для батчевой обработки данных (ETL, фичеринг) и обучения моделей на больших данных. Запуск на K8s позволяет динамически выделять ресурсы под задачу и платить только за время работы, что экономит бюджет по сравнению с постоянно работающим кластером (EMR on-demand).
Трекинг экспериментов и версионирование моделей	MLflow (Self-hosted на K8s)	Открытый, гибкий стандарт. Позволяет логировать параметры, метрики и артефакты (модель, датасет). Self-hosted развёртывание на том же K8s кластере минимизирует затраты по сравнению с SaaS-аналогами (Weights & Biases) при

		умеренном объёме экспериментов (3 ML-инженера).
Батч-инференс	Airflow Trigger -> Spark Job	Так как SLA по задержке не критичен (раз в сутки), нет необходимости в дорогом и сложном realtime-сервисе (KFServing, Seldon). Запуск батчевого Spark-задания — самый дешёвый и надёжный способ сгенерировать 10 000 прогнозов.
Мониторинг	Prometheus + Grafana (K8s stack) + Evidently AI	Prometheus/Grafana — стандарт для мониторинга инфраструктуры K8s (CPU, RAM, latency задач). Evidently AI — открытая библиотека для мониторинга дрейфа данных и качества моделей, которая легко встраивается в Airflow DAG после инференса. Альтернативы (SaaS APM) не укладываются в бюджет.

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Тема 1.3 Данные и их подготовка.

Практическое занятие №5. Сбор и базовая подготовка набора данных

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ*

Цель:

Сформировать навык поиска, первичной оценки и загрузки разнородных данных для задачи машинного обучения.

Задание:

Найдите и скачайте два открытых датасета: (а) почасовое потребление электроэнергии в одном из регионов за 3 года; (б) исторические метеоданные (температура, влажность) для этого же региона за тот же период.

Загрузите данные в рабочую среду (например, Pandas DataFrame).

Проведите первичный анализ: определите размерность, типы данных, наличие дубликатов.

Выполните базовый join двух таблиц по дате/времени.

Сохраните объединённый датасет в формате Parquet.

Пример решения:

```python
import pandas as pd

# 1. Загрузка
consumption = pd.read_csv('energy_consumption.csv', parse_dates=['datetime'])
weather = pd.read_csv('weather_data.csv', parse_dates=['datetime'])

# 2. Первичный анализ
print(consumption.info())
print(f"Дубликаты в consumption: {consumption.duplicated().sum()}")

# 3. Очистка названий колонок для join
weather.columns = [col.lower() for col in weather.columns]

# 4. Объединение
merged_df = pd.merge(consumption, weather, on='datetime', how='inner')

# 5. Проверка и сохранение
print(f"Итоговая размерность: {merged_df.shape}")
merged_df.to_parquet('energy_weather_dataset.parquet', index=False)
```

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Практическое занятие №6. Препроцессинг: очистка, нормализация, обработка пропусков.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*

- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ*

Цель:

Отработать применение стандартных техник предобработки данных с использованием scikit-learn.

Задание:

Используя датасет из предыдущего задания:

- Обработайте пропуски: если пропуск в числовом столбце (например, влажность) — заполните медианой; если в категориальном — модой.
- Нормализуйте числовые признаки (temperature, humidity, consumption) с помощью StandardScaler.
- Закодируйте категориальный признак (например, weather_condition: «солнечно», «облачно») с помощью OneHotEncoder.
- Соберите все шаги в единый sklearn Pipeline.
- Примените пайплайн к данным и сохраните трансформер (joblib.dump) для будущего использования.

Пример решения:

```python
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
import joblib

numeric_features = ['consumption', 'temperature', 'humidity']
categorical_features = ['weather_condition']

numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')),
    ('scaler', StandardScaler())
])

categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)
    ])

# Применение
processed_data = preprocessor.fit_transform(merged_df)

# Сохранение
joblib.dump(preprocessor, 'preprocessor.pkl')
```

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Практическое занятие №7. Настройка data lineage и ведение версий данных.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ*

Цель:

Научиться отслеживать происхождение данных и версионировать их с помощью специализированных инструментов.

Задание:

Установите и инициализируйте DVC (Data Version Control) в вашем проекте. Добавьте файл energy_weather_dataset.parquet под версионирование DVC. Создайте dvc.yaml, описывающий пайплайн: вход — сырые CSV, шаги — скрипты очистки и объединения, выход — финальный Parquet. Сделайте коммит в Git с файлами dvc.yaml, dvc.lock и .gitignore (где игнорируется сам Parquet). Продемонстрируйте, как команда dvc repro пересоздаёт данные, а dvc pull — скачивает их из удалённого хранилища.

Пример решения:

```
# Инициализация
dvc init
git commit -m "Initialize DVC"

# Добавление данных
dvc add data/processed/energy_weather_dataset.parquet
07:36
echo "data/processed/energy_weather_dataset.parquet" >> .gitignore
git add data/processed/energy_weather_dataset.parquet.dvc .gitignore
git commit -m "Add dataset to DVC"

# Описание пайплайна (dvc.yaml)
# stages:
#   prepare_data:
#     cmd: python scripts/prepare.py
#     deps:
#       - data/raw/consumption.csv
#       - data/raw/weather.csv
#     outs:
#       - data/processed/energy_weather_dataset.parquet
```

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Тема 1.4 Обучение, валидация и развертывание.

Практическое занятие №8. Развертывание модели в тестовой среде и базовое мониторирование.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ*

Цель:

Реализовать базовое API-развёртывание модели и настроить логирование входящих запросов.

Задание:

Обучите простую модель (например, LinearRegression) на подготовленных данных. Создайте REST API на FastAPI с одним эндпоинтом /predict. Контейнеризируйте приложение с помощью Docker. Реализуйте логирование каждого входящего запроса (признаки, время, prediction) в локальный JSONL-файл. Напишите Python-скрипт, который отправляет 5 тестовых запросов к вашему API.

Пример решения:

```python
# app/main.py (фрагмент)
from fastapi import FastAPI
import joblib
import logging
from pydantic import BaseModel

app = FastAPI()
model = joblib.load('model.pkl')
preprocessor = joblib.load('preprocessor.pkl')
logging.basicConfig(filename='api_requests.log', level=logging.INFO, format
    ='%(message)s')

class InputData(BaseModel):
    consumption: float
    temperature: float
    # ... другие поля

@app.post("/predict")
def predict(data: InputData):
    features = pd.DataFrame([data.dict()])
    processed = preprocessor.transform(features)
    prediction = model.predict(processed)[0]

    # Логирование
    logging.info(f"Input: {data.json()}, Prediction: {prediction}")

    return {"prediction": prediction}
```

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Тема 1.5 Мониторинг, обновление и управление версиями.

Практическое занятие №9. Настройка мониторинга и алертинга.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ*

Цель

Настроить сбор метрик производительности модели и системы, а также базовый алертинг.

Задание.

Разверните Prometheus и Grafana локально (например, через docker-compose). В код FastAPI из прошлого задания добавьте Prometheus-клиент для сбора: (а) количества запросов (counter); (б) времени ответа (histogram); (в) среднего значения предсказания (gauge). Экспортируйте метрики на эндпоинт /metrics. В Grafana создайте простой дашборд, отображающий RPS (requests per second) и p95 latency. Настройте в Prometheus правило: если p95 latency > 500 мс в течение 1 минуты, выставить статус CRITICAL.

Пример решения.

```python
# В код FastAPI
from prometheus_client import Counter, Histogram, Gauge, generate_latest
REQUEST_COUNT = Counter('request_count', 'Total requests')
PREDICTION_VALUE = Gauge('prediction_value', 'Last prediction value')

@app.get("/metrics")
def metrics():
    return generate_latest()

@app.post("/predict")
def predict(data: InputData):
    REQUEST_COUNT.inc()
    # ... код предсказания
    PREDICTION_VALUE.set(prediction)
    # ...
```

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Практическое занятие №10. Управление версиями: настройка реестра моделей и данных.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*

- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ*

Цель

Организовать централизованное хранилище для моделей и данных с поддержкой версионирования.

Задание.

Разверните MLflow Tracking Server локально. Модифицируйте код обучения модели: оберните его в MLflow run. Автоматически логируйте в MLflow: параметры (гиперпараметры), метрики (RMSE, MAE), артефакты (модель как **.pkl*, preprocessor.pkl, sample данных). Пометьте лучшую модель тегом production. Напишите скрипт, который загружает модель с тегом production из MLflow по её имени.

Пример решения.

```
import mlflow
import mlflow.sklearn

with mlflow.start_run(run_name="linear-regression-baseline") as run:
    mlflow.log_params(model.get_params())
    mlflow.log_metric("rmse", rmse_score)
    mlflow.sklearn.log_model(model, "model")
    mlflow.log_artifact("preprocessor.pkl")

    # Если это лучшая модель
    if rmse_score < best_score:
        mlflow.set_tag("production", "true")

# Загрузка
model_uri = "runs:/<run_id>/model"
loaded_model = mlflow.sklearn.load_model(model_uri)
```

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для

контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.

- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Практическое занятие №11. Анализ инцидента и выработка мер реагирования.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ*

Цель:

Развить навык проведения post-mortem анализа и составления плана корректирующих действий.

Задание:

Сценарий: в 14:00 система начала выдавать аномально низкие прогнозы спроса (на 40% ниже нормы). В 14:30 алерт в Grafana был замечен. В 15:00 модель

была откатана к предыдущей версии, показатели нормализовались.

Ваша задача: напишите post-mortem отчёт, включающий:

- Summary: краткое описание инцидента.
- Timeline: временная шкала событий.
- Root Cause Analysis: корневая причина (например, «в фиче is_holiday из-за сбоя внешнего API начали приходить null значения, которые после imputer стали 0»).
- Resolution: что было сделано для немедленного исправления.
- Action Items: список из 3 мер для предотвращения повторения (например, «добавить валидацию на null в пайплайн», «настроить тест на sanity-check предсказаний»).

Пример решения:

Action Items:

[Task] Добавить unit-тест для preprocessor, проверяющий, что на выходе нет NaN.
[Task] Реализовать sanity-check в API: если prediction < mean(past_24h) * 0.5, возвращать ошибку 500 и алерт.
[Process] Внедрить обязательный staging-тест с прогоном исторических данных перед релизом модели.

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Тема 1.6 Этика, безопасность и соответствие.

Практическое занятие №12. Анализ кейсов этики и приватности.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах*

Цель

Научиться выявлять и оценивать этические риски и риски приватности в ML-проектах.

Задание

Вам поручено разработать модель для скоринга кандидатов на работу на основе их видео-интервью (анализ речи, эмоций, ключевых слов). Перечислите не менее 5 потенциальных этических и правовых рисков (например, дискриминация по акценту, предвзятость к определённому полу). Для каждого риска предложите конкретную техническую или процессную меру митигации. Определите, какие данные из интервью являются PII (Personally Identifiable Information) и как их обрабатывать согласно принципам Privacy by Design.

Пример решения

Риск	Мера митигации
Дискриминация по возрасту (модель учится на мимике)	Полное удаление видеопотока, использование только ASR (Automatic Speech Recognition) текста. Предтренированная модель анализа эмоций не используется.
Предвзятость к диалектам	Использование сбалансированного датасета для дообучения ASR-модели. Регулярный аудит метрик качества (WER) по разным демографическим группам.
Хранение PII (голос, лицо)	Обработка в реальном времени без сохранения исходных медиа. Хранение только текстовой расшифровки в обезличенном виде.

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Практическое занятие №13. Разработка политики безопасности проекта.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и*

интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,

- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах*

Цель:

Сформировать базовые документы по безопасности ML-системы.

Задание:

Разработайте два документа (по 5–7 пунктов в каждом):

- Data Security Policy: политика безопасности данных. Опишите, как шифруются данные в покое (at rest) и в транзите (in transit), кто имеет доступ к сырым данным, как управляются ключи шифрования.
- Model Security Policy: политика безопасности модели. Опишите, как защищён доступ к реестру моделей (MLflow), как контролируется API-доступ к модели (токены, rate limiting), как защищается модель от кражи (model extraction attacks).

Пример решения:

Data Security Policy (фрагмент):

Все данные в S3-совместимом хранилище шифруются с помощью SSE-KMS.

Доступ к сырым данным разрешён только с IAM-ролей DataEngineer и MLEngineer через VPN.

Ключи шифрования ротируются автоматически каждые 90 дней.

<u>Критерии оценки практических заданий:</u>

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют

заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.

- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Тема 1.7 Практика интеграции, кейсы и современные тренды.

Практическое занятие №14. Интеграция ML-пайплайна: проектирование и оркестрация.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах*

Цель

Спроектировать и визуализировать полный end-to-end пайплайн обработки данных и обучения модели.

Задание

Нарисуйте схему (DAG) полного ML-пайплайна, включающего этапы: Extract, Validate Schema, Clean, Feature Engineering, Split, Train, Evaluate, Register Model, Report. Выберите инструмент оркестрации (Airflow, Dagster, Prefect) и обоснуйте выбор. Опишите 3 зависимости между задачами (например, Train зависит от Feature Engineering и Split). Опишите стратегию retry и SLA для критически важных задач.

Пример решения

DAG: extract -> validate -> clean -> features -> split -> train -> evaluate -> register -> report

Обоснование Airflow: зрелая экосистема, большое количество готовых операторов для Kubernetes, S3, MLflow. Широко известен в команде.

Зависимость: train.set_upstream([features, split]).

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Практическое занятие №15. Мультимодальные подходы и межмодельное взаимодействие.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*
- *ПК 3.12 Планировать управление требованиями заказчика в проектах*

Цель

Изучить паттерны построения систем, использующих несколько моделей и модальностей данных.

Задание

Спроектируйте систему «Умный ассистент для техподдержки». Опишите, какие модальности данных используются (текст запроса, история тикетов, скриншот ошибки). Нарисуйте схему взаимодействия моделей:

Модель 1 (NLP): классификация намерения пользователя.

Модель 2 (CV): анализ скриншота на наличие известных ошибок.

Модель 3 (Ranker): ранжирование статей KB (Knowledge Base) на основе эмбеддингов от Модели 1 и контекста от Модели 2.

Определите формат обмена данными между моделями (например, JSON с эмбеддингами и confidence score).

Пример решения

Схема: User Input (Text + Image) -> Parallel: NLP_Model(Text) & CV_Model(Image) -> [Intent, Image_Objects] -> Ranker_Model -> [Top-3 KB Articles]
Формат обмена:

```
{
  "intent": "connection_error",
  "confidence": 0.95,
  "image_analysis": {"os_error_code": "0x0000007B", "confidence": 0.88}
}
```

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора (go/no-go) и напрямую влияют на итоговый результат – 1 балл.
- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

Практическое занятие №16. Дизайн полного цикла управления моделью с документацией и презентацией.

Проверяемые результаты обучения:

- *ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам,*
- *ОК 02 Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности,*
- *ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста,*
- *ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.*
- *ПК 3.11 Создавать и управлять концепцией ИТ-продукта*

- *ПК 3.12 Планировать управление требованиями заказчика в проектах*

Цель

Синтезировать все полученные знания для описания полного жизненного цикла модели в формате, понятном для стейкхолдеров.

Задание

Подготовьте пакет документов для сдачи проекта:

- ML Model Card (1 стр.): название, задача, фреймворк, метрики качества на трейне/тесте, ограничения, дата обучения.
- Data Card (1 стр.): источники, описание фич, год сбора, известные смещения (biases), лицензия.
- Архитектурная схема (1 слайд): от данных до API.
- Краткая презентация для бизнеса (3 слайда): (1) Какую проблему решаем и сколько денег экономим; (2) Как это работает (упрощённо); (3) Как мы обеспечиваем надёжность и как будем развивать.

Пример решения

ML Model Card (фрагмент):

- Name: DemandForecast-v4.2
- Task: Time-series forecasting (14 days).
- Framework: LightGBM 4.0
- Metrics: MAE = 150 kWh, MAPE = 4.2%
- Limitations: Точность падает на 30% в период национальных праздников из-за малого количества примеров в обучающей выборке.
- Training Data: до 2025-10-01.

Критерии оценки практических заданий:

- Корректность декомпозиции: жизненный цикл модели декомпозирован на логически последовательные, общепринятые в индустрии фазы. Границы между фазами чёткие, пропуски ключевых этапов отсутствуют – 1 балл.
- Адекватность артефактов: перечисленные артефакты соответствуют заявленной фазе, являются её прямым результатом и необходимы для контроля, воспроизводимости или передачи результата на следующий этап – 1 балл.
- Существенность решений: указанные решения и действия являются критически важными для данной фазы, отражают ключевые точки выбора

(go/no-go) и напрямую влияют на итоговый результат – 1 балл.

- Целостность и связность: продемонстрировано понимание потока работ: артефакты, созданные на одной фазе, логично и корректно используются в качестве входных данных для последующих фаз – 1 балл.
- Обобщенность и системность мышления: предложенная структура решения не привязана к узкоспецифичным деталям конкретного кейса, а демонстрирует понимание универсальных принципов и шаблонов управления жизненным циклом моделей ИИ. – 1 балл.

4. Контроль приобретения практического опыта. Оценка по учебной и (или) производственной практике (по профилю специальности)

4.1. Общие положения

Целью оценки по учебной и (или) производственной практике (по профилю специальности) является оценка: 1) профессиональных и общих компетенций; 2) практического опыта и умений.

Оценка по учебной и (или) производственной практике выставляется на основании данных аттестационного листа (характеристики профессиональной деятельности обучающегося/студента на практике) с указанием видов работ, выполненных обучающимся во время практики, их объема, качества выполнения в соответствии с технологией и (или) требованиями организации, в которой проходила практика.

4.2. Виды работ практики и проверяемые результаты обучения по профессиональному модулю[1]

Таблица 8

Иметь практический опыт	**Виды и объем работ на учебной практике, требования к их выполнению и/ или условия выполнения**	**Документ, подтверждающий качество выполнения работ**
1	**2**	**3**
- Подбора и настройки готовых моделей ИИ с учетом поставленных задач, анализ результатов их применения - Создания сценариев обучения, подготовки данных для обучения, настройки гиперпараметров для достижения оптимального результата - Обучения моделей на подготовленных данных, применения методов калибровки для улучшения точности моделей - Оценки эффективности обученных моделей, корректировки обучения при необходимости, анализа ошибок и улучшения модели - Создания отчетов по обучению моделей, использования инструментов для визуализации (Matplotlib, Seaborn) для наглядного представления данных - Формирования запросов для получения и анализа данных,	1.Осуществлять выбор готовых моделей искусственного интеллекта 2.Формировать сценарии обучения готовых моделей искусственного интеллекта 3.Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта 4.Контролировать результат обучения 5.Оформлять результат проведения процедуры обучения 6.Формировать запросы для работы с искусственным интеллектом с целью визуализации данных 7.Формировать сценарии обучения готовых моделей искусственного интеллекта 8.Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов 9.Создавать контент	Аттестационный лист о прохождении практики.

построения графиков и диаграмм для визуализации результатов работы ИИ - Определения источников больших данных для анализа, идентификации внешних и внутренних источников данных для проведения аналитических работ - Получения и фильтрации больших объемов данных из гетерогенных источников - Извлечения, проверки и очистки больших объемов данных из гетерогенных источников - Агрегации и разработки представления больших объемов данных из гетерогенных источников - Оценки соответствия набора данных предметной области и задачам аналитических работ - Выбора методов и инструментальных средств анализа больших данных для проведения аналитических работ - Разработки, проверки и оценки используемых моделей больших данных - Адаптации и развертывания моделей больших данных в предметной среде - Выбора средств представления результатов аналитики больших данных - Подготовки отчета по результатам аналитических работ - Разработки кода ИС и баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Верификации кода ИС и баз данных ИС относительно дизайна ИС и структуры баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Устранения обнаруженных несоответствий в коде ИС и	программного продукта средствами нейросетей 10.Разрабатывать блоки программного кода с использованием нейросетей 11.Создавать и управлять концепцией ИТ-продукта 12.Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ	

базах данных ИС относительно дизайна ИС и структуры баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разработки интерфейсов обмена данными между ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разработки форматов обмена данными между ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разработки технологий обмена данными между ИС заказчика в соответствии с трудовым заданием в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Переработки текстов различной тематики (рерайт) - Поисковой оптимизации и адаптации текстовых материалов - Создания программного кода в соответствии с техническим заданием (готовыми спецификациями) - Оптимизации программного кода с использованием специализированных программных средств - Сбора идей о требованиях к ИТ-продукту из различных источников - Разработки функциональных требований к ИТ-продукту - Создания макетов интерфейса ИТ-продукта - Разработки нефункциональных требований к ИТ-продукту - Организации разработки и разработки плана управления требованиями заказчика в проектах в области ИТ любого		

уровня сложности - Согласования плана управления требованиями в проектах в области ИТ любого уровня сложности с заинтересованными лицами - Утверждения плана управления требованиями заказчика в проектах в области ИТ любого уровня сложности		

Таблица 9

Иметь практический опыт	**Виды и объем работ на производственной практике (по профилю специальности), требования к их выполнению и/ или условия выполнения**	**Документ, подтверждающий качество выполнения работ**
1	**2**	**3**
- Подбора и настройки готовых моделей ИИ с учетом поставленных задач, анализ результатов их применения - Создания сценариев обучения, подготовки данных для обучения, настройки гиперпараметров для достижения оптимального результата - Обучения моделей на подготовленных данных, применения методов калибровки для улучшения точности моделей - Оценки эффективности обученных моделей, корректировки обучения при необходимости, анализа ошибок и улучшения модели - Создания отчетов по обучению моделей, использования инструментов для визуализации (Matplotlib, Seaborn) для наглядного представления данных - Формирования запросов для получения и анализа данных, построения графиков и диаграмм для визуализации результатов работы ИИ - Определения источников больших данных для анализа, идентификации внешних и внутренних источников данных для	1. Анализ прикладной задачи, сравнение характеристик и выбор готовой модели искусственного интеллекта. 2. Подготовка и разделение набора данных, очистка и преобразование данных для обучения и валидации модели. 3. Разработка сценария обучения готовой модели, выбор параметров и настройка гиперпараметров. 4. Проведение обучения и калибровки модели на подготовленных данных с фиксацией параметров эксперимента. 5. Расчет и анализ метрик качества, выявление ошибок, корректировка параметров и повторная оценка модели. 6. Визуализация и оформление результатов обучения, подготовка модели к применению в информационной системе. 7. Анализ архитектуры информационной системы, бизнес-процессов и требований к использованию готовой ИИ-модели. 8. Проектирование сценария интеграции, интерфейсов обмена данными и взаимодействия ИИ-компонента с системой. 9. Подключение готовой модели	Аттестационный лист о прохождении практики, характеристика с места практики.

проведения аналитических работ - Получения и фильтрации больших объемов данных из гетерогенных источников - Извлечения, проверки и очистки больших объемов данных из гетерогенных источников - Агрегации и разработки представления больших объемов данных из гетерогенных источников - Оценки соответствия набора данных предметной области и задачам аналитических работ - Выбора методов и инструментальных средств анализа больших данных для проведения аналитических работ - Разработки, проверки и оценки используемых моделей больших данных - Адаптации и развертывания моделей больших данных в предметной среде - Выбора средств представления результатов аналитики больших данных - Подготовки отчета по результатам аналитических работ - Разработки кода ИС и баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Верификации кода ИС и баз данных ИС относительно дизайна ИС и структуры баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Устранения обнаруженных несоответствий в коде ИС и базах данных ИС относительно дизайна ИС и структуры баз данных ИС в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разработки интерфейсов обмена данными между ИС заказчика в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Разработки форматов обмена данными между ИС заказчика в рамках выполнения работ по созданию (модификации) и	через программный интерфейс, настройка передачи входных и выходных данных. 10. Разработка и настройка сценариев автоматизации бизнес-процессов с использованием результатов работы ИИ-модели. 11. Тестирование корректности интеграции, анализ качества результатов и поведения ИИ-компонента в информационной системе. 12. Оценка рисков, этических и правовых ограничений применения ИИ; документирование результатов интеграции. 13. Анализ задачи, контекста, ограничений и критериев качества ответа; формулирование требований к промпту. 14. Разработка системных и пользовательских промптов, шаблонов, примеров и правил управления контекстом. 15. Разработка цепочек и автоматизированных шаблонов промптов для выполнения типовых и многошаговых задач. 16. Создание, редактирование и адаптация текстового и иного контента программного продукта средствами нейросетей. 17. Формирование запросов для генерации, анализа, исправления и оптимизации блоков программного кода средствами нейросетей. 18. Проведение A/B-тестирования промптов, расчет метрик качества, версионирование и улучшение промптов с учетом безопасности. 19. Сбор, анализ и приоритизация требований к ИТ-продукту и ИИ-компоненту; формирование концепции решения. 20. Разработка плана жизненного цикла модели, архитектуры решения, этапов работ и критериев приемки результата. 21. Организация подготовки данных, обучения, валидации и развертывания модели в соответствии с планом проекта. 22. Настройка учета экспериментов, версий моделей и артефактов,	

сопровождению ИС - Разработки технологий обмена данными между ИС заказчика в соответствии с трудовым заданием в рамках выполнения работ по созданию (модификации) и сопровождению ИС - Переработки текстов различной тематики (рерайт) - Поисковой оптимизации и адаптации текстовых материалов - Создания программного кода в соответствии с техническим заданием (готовыми спецификациями) - Оптимизации программного кода с использованием специализированных программных средств - Сбора идей о требованиях к ИТ-продукту из различных источников - Разработки функциональных требований к ИТ-продукту - Создания макетов интерфейса ИТ-продукта - Разработки нефункциональных требований к ИТ-продукту - Организации разработки и разработки плана управления требованиями заказчика в проектах в области ИТ любого уровня сложности - Согласования плана управления требованиями в проектах в области ИТ любого уровня сложности с заинтересованными лицами - Утверждения плана управления требованиями заказчика в проектах в области ИТ любого уровня сложности	обеспечение воспроизводимости и процедуры обновления. 23. Организация мониторинга качества и стабильности модели, выявление деградации и планирование обновления или отката. 24. Управление требованиями, рисками, безопасностью и соответствием; оформление документации и взаимодействие с заинтересованными сторонами.	

5. Контрольно-оценочные материалы для квалификационного экзамена

СВОДНАЯ ВЕДОМОСТЬ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ
ПМ.03 Обучение готовых моделей искусственного интеллекта
Ф.И.О.«_____»
студент на 4 курсе по специальности СПО 09.02.13 «Интеграция решений с применением технологий искусственного интеллекта»
освоил программу профессионального модуля **ПМ.03** Обучение готовых моделей искусственного интеллекта
в объёме 1036 час. с «____» _____20__ г. по «___»_____ 20__г.

Результаты промежуточной аттестации по элементам профессионального модуля

Элементы модуля	**Формы промежуточной аттестации**	**Оценка**
МДК 03.01 Разработка сценариев обучения готовых моделей	Экзамен	
МДК 03.01 Разработка сценариев обучения готовых моделей (курсовой проект)	Дифференцированный зачет	
МДК 03.02 Интеграция искусственного интеллекта в информационные системы	Дифференцированный зачет	
МДК 03.03 Разработка промптов для искусственного интеллекта	Дифференцированный зачет	
МДК 03.04 Управление жизненным циклом моделей ИИ	Дифференцированный зачет	
УП.03	Зачет	
ПП.03	Зачет	
ПМ 03	**Экзамен по модулю**	

Итоги экзамена по профессиональному модулю

Коды проверяемых компетенций	**Показатели оценки результата**	**Оценка (освоен / не освоен)**
ОК 01	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	освоен
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности	освоен
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	освоен
ОК 09	Пользоваться профессиональной документацией на государственном и иностранном языках	освоен
ПК 3.1	Осуществлять выбор готовых моделей искусственного интеллекта	освоен
ПК 3.2	Формировать сценарии обучения готовых моделей искусственного интеллекта	освоен
ПК 3.3	Проводить обучение и последующую калибровку готовых моделей искусственного интеллекта	освоен
ПК 3.4	Контролировать результат обучения	освоен

ПК 3.5	Оформлять результат проведения процедуры обучения	освоен
ПК 3.6	Формировать запросы для работы с искусственным интеллектом с целью визуализации данных	освоен
ПК 3.7	Формировать сценарии обучения готовых моделей искусственного интеллекта	освоен
ПК 3.8	Интегрировать готовые модели искусственного интеллекта в информационные системы с использованием программных интерфейсов	освоен
ПК 3.9	Создавать контент программного продукта средствами нейросетей	освоен
ПК 3.10	Разрабатывать блоки программного кода с использованием нейросетей	освоен
ПК 3.11	Создавать и управлять концепцией ИТ-продукта	освоен
ПК 3.12	Планировать управление требованиями заказчика в проектах любого уровня сложности в области ИТ	освоен
Члены аттестационной комиссии _____ _____		

Лист согласования

Дополнения и изменения к комплекту ФОС по ПМ.__
на_____ учебный год

_____

Дополнения и изменения к комплекту ФОС на _____ учебный год по профессиональному модулю _____

В комплект ФОС внесены следующие изменения:

_____

_____

_____

_____

_____

Дополнения и изменения в комплекте ФОС обсуждены на заседании ЦК _____Протокол № _____ от «____» _____ 20____г.

Председатель ЦК _____