

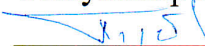
Федеральное государственное образовательное
бюджетное учреждение высшего образования
**«Финансовый университет при Правительстве
Российской Федерации»
(Финансовый университет)**

Владикавказский филиал Финуниверситета

УТВЕРЖДАЮ

Директор

Владикавказского филиала
Финуниверситета

 Т.А. Хубаев

«31» августа 2026 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по профессиональному модулю

ПМ.01 Разработка кода для обучения искусственного интеллекта

09.02.13 Интеграция решений с применением технологий искусственного
интеллекта

Владикавказ 2026

Фонд оценочных средств по профессиональному модулю разработан на основе федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.13 Интеграция решений с применением технологий искусственного интеллекта.

Составитель:

Зембатова М.А., преподаватель

Фонд оценочных средств по профессиональному модулю рассмотрен и рекомендован к утверждению на заседании предметной (цикловой) комиссии математики и информатики

Протокол от «28» 08 2026 г. № 1

Председатель предметной (цикловой)
комиссии математики и информатики

 М.К. Ходова

Согласовано:

Ковалева М.А., начальник отдела научных исследований и образовательной деятельности ООО «Экспертно-аналитические системы»

1. Паспорт фонда оценочных средств
по профессиональному модулю ПМ.01 Разработка кода для обучения
искусственного интеллекта
09.02.13 Интеграция решений с применением технологий искусственного
интеллекта

Результаты обучения (практический опыт, знания, умения)	Общие и профессионал ьные компетенции	Наименование элементов профессиональ ного модуля, раздела, темы	Наименование оценочного средства	
			Текущий контроль успеваемо сти	Промежуточная аттестация
Раздел 1. Разработка программных модулей в системах искусственного интеллекта				
МДК 01.01. Разработка программных модулей в системах ИИ				
Практический опыт: – разработки, оптимизации и оценки сложности алгоритмов для ИИ-программ; – применения структур данных; – разработки модульных ИИ-систем; – внедрения разработанных ИИ-модулей; – оптимизации кода и работы с интерфейсами; – оформления и документирования кода; – управления проектами с использованием VCS; – настройки процессов CI/CD; отладки программных модулей. Знания: – основные методы и подходы к построению алгоритмов; – языки программирования. принципы модульного программирования; – стандартные фреймворки и	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 04 Эффективно взаимодействовать и работать в коллективе и команде	Тема 1.1. Введение в искусственный интеллект и его направления Тема 1.2. Методы сбора и предобработки данных Тема 1.3. Основы алгоритмов машинного обучения Тема 1.4. Оценка качества моделей и улучшение алгоритмов Тема 1.5. Глубокое обучение и нейронные сети Тема 1.6. Проектирование ИИ-систем	– вопросы для устного и/или письменного опроса; – практические задания; – задания для лабораторных работ; – кейс-задания; – тестовые задания; – тематика рефератов, докладов, сообщений.	Экзаменационные вопросы и задания для устного (письменного) экзамена по междисциплинарному курсу Экзаменационные вопросы и задания для устного (письменного) экзамена по модулю

библиотеки для работы с ИИ; – принципы чистого кода; – инструменты для автоматической проверки качества кода; – принципы работы VCS; – принципы работы отладчиков и логирования. Умения: – анализировать технические задания и выявлять требования к алгоритмам; – применять методы алгоритмизации; – разрабатывать оптимальные алгоритмы для задач ИИ; – реализовывать программные модули. – соблюдать принципы «чистого кода»; – использовать стандартные библиотеки и фреймворки; – оформлять и документировать код; – работать с системами контроля версий; – использовать инструменты для отладки.	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 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием. ПК 1.3. Оформлять программный код в соответствии с техническим заданием. ПК 1.4. Использовать систему	Учебная практика раздела 1 Производственная практика (по профилю специальности) раздела 1	– виды работ и задания по учебной практике – виды работ и задания по производственной практике (по профилю специальности)	Документы, подтверждающие выполнение соответствующих работ по учебной практике (отчет по практике, индивидуальное задание по практике, дневник по практике, характеристика-отзыв о прохождении практики, аттестационный лист с оценкой уровня сформированности компетенций) Вопросы для защиты отчета по учебной практике Документы, подтверждающие выполнение соответствующих работ по производственной практике (по профилю специальности) (отчет по практике, индивидуальное задание по практике, дневник по практике, характеристика-отзыв о прохождении практики, аттестационный лист с оценкой уровня сформированности компетенций) Вопросы для защиты отчета по производственной практике (по профилю специальности)
--	---	---	---	---

	контроля версий программного кода с учетом обеспечения возможности организации групповой разработки. ПК 1.5. Выполнять отладку программных модулей с использованием специализированных программных средств			
Раздел 2. Разработка мобильных приложений с поддержкой искусственного интеллекта				
МДК 01.02. Разработка мобильных приложений с поддержкой ИИ				
Практический опыт: – разработки модульных ИИ-систем, соответствующих требованиям производительности и безопасности; – оптимизации кода и работы с интерфейсами – использования инструментов статического анализа кода; – работы с системами документирования кода; – разрешения конфликтов при слиянии веток и использования pull request для рецензирования кода.	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	Тема 2.1. Платформы и инструменты мобильной разработки Тема 2.2. Интеграция ИИ в мобильные приложения Тема 2.3. Разработка интерактивных мобильных ИИ-приложений Тема 2.4. Развертывание мобильных приложений с ИИ	– вопросы для устного и/или письменного опроса; – практические задания; – задания для лабораторных работ; – кейс-задания; – тестовые задания; – тематика рефератов, докладов, сообщений	Вопросы для устного (письменного) зачета по междисциплинарному курсу Экзаменационные вопросы и задания для устного (письменного) экзамена по модулю

Знания: – стандартные фреймворки и библиотеки; – принципы работы распределенных систем контроля версий; – основные команды и операции в VCS; – методы разрешения конфликтов в ходе групповой разработки; – инструменты для отладки кода; – способы выявления ошибок в программе. Умения: – реализовывать программные модули на основе требований ТЗ; – соблюдать соглашения о наименованиях; – организовывать совместную работу над проектом через ветки разработки; – разрешать конфликты при слиянии кода; – применять методы логирования для анализа выполнения программ.	задач профессиональной деятельности. ОК 04 Эффективно взаимодействовать и работать в коллективе и команде 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 1.1.	Тема 2.5. Безопасность и защита данных в мобильных ИИ-приложениях Тема 2.6. Проектная деятельность и сопровождение мобильных ИИ-приложений Учебная практика раздела 2	й. – виды работ и задания по учебной практике	Документы, подтверждающие выполнение соответствующих работ по учебной практике (отчет по практике, индивидуальное задание по практике, дневник по практике, характеристика-отзыв о прохождении практики, аттестационный лист с оценкой уровня сформированности компетенций) Вопросы для защиты отчета по учебной практике
---	--	--	---	--

	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием. ПК 1.3. Оформлять программный код в соответствии с техническим заданием. ПК 1.4. Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки. ПК 1.5. Выполнять отладку программных модулей с использованием специализированных программных средств	Производственная практика (по профилю специальности) раздела 2	– виды работ и задания по производственной практике (по профилю специальности)	Документы, подтверждающие выполнение соответствующих работ по производственной практике (по профилю специальности) (отчет по практике, индивидуальное задание по практике, дневник по практике, характеристика-отзыв о прохождении практики, аттестационный лист с оценкой уровня сформированности компетенций) Вопросы для защиты отчета по производственной практике (по профилю специальности)
Раздел 3. Тестирование программных модулей				
МДК 01.03. Тестирование программных модулей				

Практический опыт: – выполнения статического тестирования; – выполнения тестирования в соответствии с тест-планом; – генерирования тестовых данных; – выполнения интеграционного и регрессионного тестирования; – работы с CI/CD пайплайнами для автоматизации тестирования; – разработки тестовых сценариев и пакетов; – оценки тестовых данных на предмет покрытия строк и ветвей; – автоматизации создания и выполнения тестов. Знания: – техники выполнения тестовых прогонов. инструменты и среды выполнения тестирования; – языки разработки автоматизированных тестов; – правила выполнения отчетов о тестировании; – цели, задачи и виды тестирования; – жизненный цикл дефекта; – основы тест-дизайна; – основные инструменты проектирования тестов. Умения: – проводить различные виды тестирования; – выполнять настройки окружения и подготовку	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 04 Эффективно взаимодействовать и работать в коллективе и команде ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста ОК 09. Пользоваться профессиональной	Тема 3.1. Основы тестирования программных приложений Тема 3.2. Планирование тестирования и проектирование артефактов Тема 3.3. Спецификационные методы тест-дизайна Тема 3.4. Особенности тестирования ИИ-систем Тема 3.5. Автоматизация тестирования ИИ-систем Тема 3.6. Тестирование ИИ-приложений Учебная практика раздела 3	– вопросы для устного и/или письменного опроса; – практические задания; – задания для лабораторных работ; – кейс-задания; – тестовые задания; – тематика рефератов, докладов, сообщений. – виды работ и задания по учебной практике –	Вопросы для устного (письменного) зачета по междисциплинарному курсу Экзаменационные вопросы и задания для устного (письменного) экзамена по модулю Документы, подтверждающие выполнение соответствующих работ по учебной практике (отчет по практике, индивидуальное задание по практике, дневник по практике, характеристика-отзыв о прохождении практики, аттестационный лист с оценкой уровня сформированности компетенций) Вопросы для защиты отчета по учебной практике
--	--	--	---	---

тестовых данных; – фиксировать результаты выполнения тестов; – определять уровень критичности дефектов; – разрабатывать автоматизированные тесты; – проектировать тестовые сценарии на основе тестовых планов; – оценивать риски при отборе тестов.	документацией на государственном и иностранном языках. ПК 1.6. Выполнять тестирование программного кода. ПК 1.7. Составлять тестовые сценарии.	Производственная практика (по профилю специальности) раздела 3	– виды работ и задания по производственной практике (по профилю специальности) –	Документы, подтверждающие выполнение соответствующих работ по производственной практике (отчет по практике, индивидуальное задание по практике, дневник по практике, характеристика-отзыв о прохождении практики, аттестационный лист с оценкой уровня сформированности компетенций) Вопросы для защиты отчета по производственной практике (по профилю специальности)
--	---	---	--	---

2. Формы промежуточной аттестации по профессиональному модулю

Элементы профессионального модуля	Формы промежуточной аттестации					
	1 семестр	2 семестр	3 семестр	4 семестр	5 семестр	6 семестр
Раздел 1 МДК 01.01. Разработка программных модулей в системах искусственного	-	-	Экзамен	-	-	-
Раздел 2 МДК 01.02. Разработка мобильных приложений с поддержкой искусственного	-	-	Дифференцированный зачет	-	-	-
Раздел 3 МДК 01.03. Тестирование программных модулей	-	-	Дифференцированный зачет	-	-	-
Учебная практика	-	-	Дифференцированный зачет	-	-	-
Производственная практика (по профилю специальности)	-	-	Дифференцированный зачет	-	-	-
ПМ.01 Разработка кода для обучения искусственного интеллекта	-	-	Экзамен по модулю	-	--	

3. Комплект оценочных средств

1. Задания для текущего контроля успеваемости

Вопросы для устного или письменного опроса

МДК 01.01. Разработка программных модулей в системах искусственного интеллекта

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

1. Дайте определение искусственного интеллекта и назовите основные исторические этапы его развития.
2. Перечислите и охарактеризуйте основные современные направления искусственного интеллекта.
3. Опишите этические проблемы и вызовы, связанные с разработкой и внедрением систем ИИ.
4. Объясните важность качества данных для обучения ИИ-моделей и назовите основные методы их сбора.
5. Опишите процесс предобработки данных, включая методы очистки, нормализации, кодирования категориальных данных и работы с пропусками и выбросами.
6. Дайте определение feature engineering и приведите примеры создания новых признаков для моделей.
7. Опишите методы валидации, контроля качества и работы с несбалансированными данными при подготовке датасетов.
8. Сравните обучение с учителем, обучение без учителя и обучение с подкреплением, приведите примеры алгоритмов для каждого вида.
9. Опишите принцип работы и области применения метода k-ближайших соседей (kNN), метода опорных векторов (SVM) и наивного байесовского классификатора.
10. Объясните разницу между линейной и логистической регрессией и приведите примеры задач для их решения.
11. Опишите процесс построения и анализа деревьев решений, а также принципы работы ансамблевых методов (случайный лес, градиентный бустинг).
12. Дайте определение кластеризации и сравните алгоритмы k-means и иерархической (агломеративной) кластеризации.
13. Опишите назначение и принцип работы метода главных компонент (PCA) для уменьшения размерности данных.
14. Опишите принцип работы алгоритма Apriori для поиска ассоциативных

- правил и приведите пример построения системы рекомендаций.
15. Перечислите основные метрики оценки качества моделей классификации и объясните их физический смысл.
 16. Опишите процесс построения и анализа матрицы ошибок (Confusion Matrix), ROC-кривой, AUC и кривых Precision-Recall.
 17. Объясните суть кросс-валидации и опишите разницу между стандартной, стратифицированной, вложенной кросс-валидацией и валидацией на временных рядах.
 18. Дайте определение регуляризации моделей и сравните L1 и L2-регуляризацию, опишите методы оптимизации гиперпараметров.
 19. Дайте определение нейронной сети и опишите архитектуру многослойного перцептрона (MLP).
 20. Объясните процесс обучения нейронных сетей, включая метод обратного распространения ошибки, стохастический градиентный спуск и пакетную нормализацию.
 21. Опишите назначение и принцип работы основных функций активации (ReLU, сигмоидальная) и современных оптимизаторов.
 22. Сравните сверточные (CNN) и рекуррентные (RNN, LSTM) нейронные сети, приведите примеры задач для их применения.
 23. Опишите назначение автоэнкодеров для сжатия данных и механизма аугментации данных в глубоком обучении.
 24. Дайте определение трансферному обучению (Transfer learning) и опишите процесс сохранения и развертывания обученной нейросети.
 25. Перечислите основные принципы проектирования архитектуры ИИ-систем (модульность, масштабируемость, эффективность).
 26. Опишите процесс контейнеризации ИИ-модели с использованием Docker и ее развертывания в Kubernetes.
 27. Объясните принципы построения микросервисной архитектуры для ИИ-сервисов и проектирования REST API.
 28. Опишите назначение CI/CD пайплайнов в машинном обучении и роль инструментов версионирования моделей и данных (MLflow, DVC).
 29. Перечислите методы обеспечения безопасности, мониторинга и надежности ИИ-систем в промышленной эксплуатации, опишите стратегии резервного копирования.
 30. Опишите подходы к интеграции очередей сообщений (RabbitMQ/Kafka), оптимизации ресурсов в облачной инфраструктуре, а также методы нагрузочного и А/В тестирования ИИ-моделей.

МДК 01.02. Разработка мобильных приложений с поддержкой
искусственного интеллекта

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

1. Дайте сравнительную характеристику платформ мобильной разработки Android и iOS, опишите их основные архитектурные отличия и инструменты создания приложений.
2. Опишите процесс установки и настройки среды разработки Android Studio, а также структуру стандартного Android-проекта и конфигурации Gradle.
3. Перечислите основные языки программирования для мобильной разработки и охарактеризуйте синтаксические особенности и преимущества Kotlin и Java.
4. Объясните принципы адаптивной верстки и опишите работу с ресурсами приложения (строки, цвета, стили, темы) и макетами (ConstraintLayout).
5. Опишите назначение и особенности работы основных UI-компонентов (RecyclerView, CardView, Navigation Component) и механизмы обработки событий взаимодействия с пользователем.
6. Перечислите инструменты отладки и профилирования Android-приложений и опишите процесс их использования (Logcat, Debugger, Layout Inspector).
7. Дайте определение фреймворку TensorFlow Lite и опишите пошаговый процесс конвертации и внедрения базовой модели ИИ в мобильное приложение.
8. Объясните принципы использования предобученных моделей ИИ для решения задач компьютерного зрения, оптического распознавания символов (OCR) и обработки речи на мобильных устройствах.
9. Опишите технический процесс интеграции камеры (CameraX) для непрерывной передачи видеопотока в ИИ-модель в реальном времени.
10. Сравните методы оптимизации ИИ-моделей для мобильных платформ, подробно объясните математическую и физическую суть квантизации и прунинга.
11. Опишите механизмы аппаратного ускорения инференса (GPU Delegate, NNAPI) и их влияние на производительность, потребление памяти и заряд батареи мобильного устройства.
12. Объясните разницу между локальным (Edge AI) и облачным выполнением ИИ-моделей и спроектируйте архитектуру гибридного мобильного приложения (онлайн/офлайн).
13. Перечислите подходы к разработке интуитивного пользовательского интерфейса и реализации мгновенной обратной связи для интерактивной

ИИ-приложений.

14. Опишите процесс разработки приложения с использованием технологий дополненной реальности (AR) и интеграции различных сенсоров устройства для сбора контекстных данных.
15. Объясните архитектурные принципы создания голосового помощника и механизмы обработки естественного языка (NLP) непосредственно в мобильном приложении.
16. Опишите полный жизненный цикл мобильного ИИ-приложения и объясните роль систем контроля версий (Git, GitFlow, Code Review) в обеспечении качества командной разработки.
17. Перечислите этапы подготовки мобильного приложения к релизу и опишите процессы сборки, оптимизации итогового размера (APK/AAB) и криптографической подписи.
18. Объясните различия между процессами публикации приложения в Google Play Console и Apple App Store, опишите организацию закрытого и открытого бета-тестирования.
19. Опишите методы пост-релизного мониторинга приложения с использованием систем аналитики (Crashlytics, ANR) и подходы к сбору обратной связи от пользователей.
20. Перечислите основные угрозы безопасности мобильных приложений и опишите механизмы реализации аутентификации, авторизации и биометрической защиты.
21. Объясните методы шифрования локальных данных, защиты сетевого трафика и безопасного хранения конфиденциальных API-ключей в коде приложения.
22. Опишите специфические уязвимости ИИ-моделей на мобильных устройствах и предложите методы защиты интеллектуальной собственности (моделей) от кражи и реверс-инжиниринга.
23. Дайте определение E2E-тестированию и опишите процесс настройки автоматизации тестирования мобильного приложения в рамках CI/CD пайплайна.
24. Объясните принципы проектирования пользовательского опыта (UX/UI) для ИИ-функций и опишите процесс локализации приложения для выхода на международные рынки.
25. Опишите методологию проведения A/B тестирования новых функций приложения и подходы к анализу метрик качества для принятия решений об обновлениях.
26. Перечислите стратегии масштабирования архитектуры мобильного ИИ-приложения и опишите процессы поддержки, выпуска исправлений и

управления жизненным циклом продукта.

МДК 01.03. Тестирование программных модулей
(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.6, ПК 1.7)

1. Дайте определение качества программного обеспечения в соответствии с ГОСТ Р ИСО/МЭК 25051 и перечислите основные метрики его оценки.
2. Опишите цели, задачи и основные уровни тестирования программного обеспечения (модульное, интеграционное, системное, приемочное).
3. Объясните разницу между функциональным и нефункциональным тестированием, приведите примеры видов тестирования производительности.
4. Опишите жизненный цикл дефекта и критерии оценки его серьезности (severity) и приоритета (priority).
5. Дайте определение тестовой стратегии и тестового плана, опишите их структуру и основные разделы.
6. Объясните назначение чек-листов, тестовых пакетов и заданий на тестирование в процессе планирования качества.
7. Опишите процесс оценки покрытия тестами бизнес-требований и функциональных спецификаций.
8. Опишите суть и правила применения метода эквивалентного разбиения и анализа граничных значений при проектировании тестов.
9. Объясните принцип построения и использования таблиц решений (decision tables) для тестирования сложной бизнес-логики.
10. Опишите технику тестирования переходов состояний (state transition testing) и приведите пример ее практического применения.
11. Дайте определение диаграмме причинно-следственных связей и опишите процесс ее использования для генерации тестовых сценариев.
12. Объясните концепцию покрытия кода (code coverage) и разницу между покрытием строк и покрытием ветвей.
13. Опишите структуру матрицы ошибок (Confusion Matrix) и дайте определение метрикам Accuracy, Precision, Recall и F1-score.
14. Объясните физический смысл ROC-кривой и метрики AUC при оценке качества моделей машинного обучения.
15. Перечислите основные метрики оценки моделей регрессии (MAE, RMSE, MAPE) и опишите, в каких случаях целесообразно использовать каждую из них.
16. Опишите проблему компромисса между смещением и дисперсией (Bias-Variance Tradeoff) и методы оценки оптимальной сложности модели ИИ.

- 17.Объясните особенности тестирования моделей кластеризации и назовите основные метрики качества для задач обучения без учителя.
- 18.Опишите критерии выбора инструментов для автоматизации тестирования и этапы внедрения автоматизированного тестирования в проект.
- 19.Объясните назначение моков (mocks) и стабов (stubs) при тестировании ИИ-систем с внешними зависимостями и API.
- 20.Опишите процесс параметризации тестов и генерации синтетических тестовых данных для моделей машинного обучения.
- 21.Объясните роль автоматизации тестирования в CI/CD пайплайнах и опишите процесс настройки генерации отчетов о тестировании.
- 22.Дайте определение E2E (End-to-End) тестированию и опишите процесс настройки тестового окружения с использованием заглушек.
- 23.Опишите особенности кросс-платформенного тестирования ИИ-интерфейсов и мобильных ИИ-приложений.
- 24.Объясните принципы проведения А/В тестирования ИИ-моделей в production-среде и подходы к анализу бизнес-метрик.
- 25.Опишите методы нагрузочного тестирования и оценки масштабируемости ИИ-систем, а также инструменты профилирования производительности.
- 26.Перечислите основные угрозы безопасности ИИ-приложений и опишите подходы к тестированию на проникновение (pentest) и отказоустойчивости.

Критерии оценки ответов студента в процессе устного и/или письменного опроса

Оценка «отлично» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания вопросов программного материала.

Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе некоторые неточности.

Оценка «удовлетворительно» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом показавшему владение основными понятиями, выносимых на рассмотрение вопросов, необходимыми для дальнейшего обучения и способность применять полученные знания по образцу в стандартной ситуации.

Оценки «неудовлетворительно» заслуживает студент, который не знает большей части основного содержания выносимых на рассмотрение вопросов

дисциплины (темы), допускает грубые ошибки в формулировках основных понятий.

Тематика рефератов, докладов, сообщений

МДК 01.01. Разработка программных модулей в системах искусственного интеллекта

((ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

1. История развития искусственного интеллекта: от Тьюринга до наших дней.
2. Этические проблемы и вызовы современного ИИ: предвзятость, приватность, безопасность.
3. Применение ИИ в различных отраслях: медицина, финансы, транспорт, образование.
4. Основные направления ИИ: сравнительный анализ машинного и глубокого обучения.
5. Методы веб-скрапинга: правовые и технические аспекты.
6. Feature engineering: искусство создания признаков для моделей.
7. Обучение с учителем, без учителя и с подкреплением: сравнительный анализ.
8. Проблема переобучения и методы борьбы с ним.
9. Трансформеры и внимание: революция в обработке естественного языка.
10. Безопасность ИИ-систем: уязвимости и методы защиты.

МДК 01.02. Разработка мобильных приложений с поддержкой искусственного интеллекта

((ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

1. Сравнительный анализ платформ мобильной разработки: Android vs iOS.
2. Локализация приложений: адаптация для разных рынков.
3. Монетизация мобильных приложений: модели и стратегии.
4. Сравнительный анализ фреймворков для ИИ на мобильных устройствах.
5. Edge AI: запуск нейронных сетей на мобильных устройствах.
6. Тенденции развития мобильной ИИ-разработки в 2024-2025.
7. Этические аспекты внедрения ИИ в мобильные приложения.
8. Оптимизация потребления батареи в ИИ-приложениях.
9. Offline-first архитектура для мобильной ИИ-приложений.
10. Прогрессивные веб-приложения (PWA) с ИИ-функционалом.
11. Будущее мобильной разработки: тренды и технологии.

МДК 01.03. Тестирование программных модулей
(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.6, ПК 1.7)

1. Чем тестирование ИИ отличается от тестирования традиционного ПО.
2. Дрейф данных (Data Drift): мониторинг и предотвращение.
3. Тестирование пайплайнов данных: особенности и методы.
4. Моки и стабы в тестировании ИИ-систем.
5. Автоматизация тестирования API для ИИ-моделей.
6. Парадокс автоматизации: когда автоматизация неэффективна.
7. E2E тестирование: полный цикл проверки приложения.
8. Сравнительный анализ инструментов тестирования ИИ-моделей.
9. Автоматическая генерация тестов для нейронных сетей.
10. Тестирование рекомендательных систем: метрики и подходы.
11. Тестирование NLP-моделей: особенности обработки естественного языка.
12. Тестирование компьютерного зрения: методы валидации.
13. Роль тестировщика в команде разработки ИИ-продуктов.
14. Тестирование голосовых помощников и чат-ботов.
15. Перспективы развития автоматизированного тестирования ИИ»

Критерии оценки рефератов, докладов, сообщений

Оценка «отлично» выставляется студенту, если содержание доклада (реферата, сообщения) соответствует тематике; доклад (реферат, сообщение) подготовлен в соответствии с общими требованиями: имеет чёткую композицию и структуру; в содержании доклада (реферата, сообщения) отсутствуют логические нарушения в представлении материала; доклад (реферат, сообщение) представляет собой самостоятельное исследование, в котором проведен качественный анализ собранного материала. В случае письменного доклада (реферата, сообщения), к тому же соблюдены технические требования: корректно оформлены и в полном объёме представлен список использованных источников и ссылки на него в тексте доклада; отсутствуют орфографические, пунктуационные, грамматические, лексические, стилистические и иные ошибки в авторском тексте.

Оценка «хорошо» выставляется студенту, если содержание доклада (реферата, сообщения) соответствует заявленной тематике; доклад (реферат, сообщение) имеет чёткую композицию и структуру; в содержании доклада (реферата, сообщения) отсутствуют логические нарушения в представлении материала; доклад (реферат, сообщение) представляет собой самостоятельное исследование, в котором проведен качественный анализ собранного материала. В случае письменного доклада (реферата, сообщения), к тому же

соблюдены технические требования, но есть погрешности в оформлении, например, не корректно оформлены и в не полном объёме представлены ссылки на использованные источники в тексте доклада (реферата, сообщения), есть единичные пунктуационные, стилистические и иные ошибки в авторском тексте.

Оценка «удовлетворительно» выставляется студенту, если содержание доклада (реферата, сообщения) соответствует заявленной тематике; в целом доклад (реферат, сообщение) представляет собой самостоятельное исследование, однако в нем представлен недостаточно полный анализ найденного материала; доклад (реферат, сообщение) имеет недостаточно чёткую композицию и структуру и/или в тексте доклада (реферата, сообщения) есть логические нарушения в представлении материала. В случае письменного оформления доклада (реферата, сообщения), есть погрешности в техническом оформлении, например, некорректно оформлены или не в полном объёме представлены ссылки на использованные источники в тексте доклада (реферата, сообщения); есть единичные орфографические, пунктуационные, грамматические, лексические, стилистические и иные ошибки в авторском тексте.

Оценки «не удовлетворительно» заслуживает студент, если в докладе отмечены нарушения общих требований его подготовки; доклад (реферат, сообщение) не представляет собой самостоятельного исследования, в содержании доклада (реферата, сообщения) есть логические нарушения в представлении материала; отсутствует анализ собранного материала, текст доклада (реферата, сообщения) представляет собой непереработанный текст другого автора (других авторов). В случае письменного оформления доклада (реферата, сообщения), есть погрешности в техническом оформлении: не в полном объёме представлен список использованных источников, есть ошибки в оформлении; некорректно оформлены или не в полном объёме представлены ссылки на использованные источники в тексте доклада (реферата, сообщения); есть частые орфографические, пунктуационные, грамматические, лексические, стилистические и иные ошибки в авторском тексте.

Тестовые задания

МДК 01.01. Разработка программных модулей в системах искусственного интеллекта

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

1. В чем заключается основная задача обучения с учителем (supervised learning)?

- a) поиск скрытых паттернов и группировка неразмеченных данных;
 - b) прогнозирование выходных значений, на основе предоставленных размеченных данных (пар «признак-ответ»);
 - c) максимизация функции вознаграждения в процессе взаимодействия со средой;
 - d) снижение размерности данных для их визуализации.
2. Какой метод предобработки данных используется для преобразования категориальных признаков в числовые путем создания отдельных бинарных столбцов?
- a) нормализация (Min-Max scaling);
 - b) One-Hot Encoding (One-Hot кодирование);
 - c) метод главных компонент (PCA);
 - d) K-fold кросс-валидация.
3. Какой алгоритм машинного обучения основан на поиске оптимальной разделяющей гиперплоскости для классификации данных?
- a) метод k-ближайших соседей (kNN);
 - b) дерево решений;
 - c) метод опорных векторов (SVM);
 - d) алгоритм Apriori.
4. Что представляет собой F1-мера (F-мера) при оценке качества модели классификации?
- a) арифметическое среднее между точностью (Precision) и полнотой (Recall);
 - b) гармоническое среднее между точностью (Precision) и полнотой (Recall);
 - c) отношение верно предсказанных положительных классов ко всем реально положительным;
 - d) площадь под ROC-кривой.
5. Какова основная цель применения k-блочной кросс-валидации (k-fold cross-validation)?
- a) ускорение процесса обучения модели на больших данных;
 - b) уменьшение переобучения и более надежная оценка обобщающей способности модели;
 - c) автоматическое заполнение пропущенных значений в датасете;
 - d) прямая оптимизация гиперпараметров без использования сетки поиска.
6. Какой тип регуляризации добавляет к функции потерь сумму абсолютных значений весов модели, способствуя обнулению незначимых признаков (Lasso)?

- a) L1-регуляризация;
 - b) L2-регуляризация (Ridge);
 - c) Dropout (Отсев);
 - d) Batch Normalization (Пакетная нормализация).
7. Какая архитектура нейронных сетей лучше всего подходит для задач компьютерного зрения и распознавания изображений?
- a) рекуррентные нейронные сети (RNN);
 - b) сверточные нейронные сети (CNN);
 - c) автоэнкодеры;
 - d) полностью связанные сети (MLP) без скрытых слоев.
8. Какую главную проблему стандартных рекуррентных нейронных сетей (RNN) призваны решить архитектуры LSTM?
- a) проблема исчезающего и взрывающегося градиента при обучении на длинных последовательностях;
 - b) неспособность работать с табличными данными;
 - c) высокая вычислительная сложность при кластеризации;
 - d) отсутствие функции активации.
9. Каково основное назначение функции активации ReLU в глубоком обучении?
- a) нормализация входных данных к диапазону $[0, 1]$;
 - b) введение нелинейности в модель, позволяя ей обучаться сложным зависимостям;
 - c) вычисление значения функции потерь (loss);
 - d) уменьшение размерности тензора.
10. Какой инструмент преимущественно используется для контейнеризации ИИ-моделей и обеспечения идентичности среды их выполнения на разных машинах?
- a) Git;
 - b) Docker;
 - c) RabbitMQ;
 - d) MLflow.
11. Какова основная роль инструмента MLflow в жизненном цикле машинного обучения (MLOps)?
- a) веб-скрапинг данных из открытых источников;
 - b) версионирование моделей, отслеживание экспериментов и управление артефактами;
 - c) проектирование пользовательских интерфейсов (UI);
 - d) ручная разметка изображений для обучения.

12. Что такое веб-скрапинг (web scraping) в контексте подготовки данных для ИИ?
- a) процесс очистки данных от дубликатов и выбросов;
 - b) автоматизированный сбор и извлечение информации с веб-сайтов;
 - c) процесс шифрования данных перед передачей в облако;
 - d) метод аугментации (расширения) изображений.
13. Какой из перечисленных алгоритмов относится к обучению без учителя (unsupervised learning)?
- a) логистическая регрессия;
 - b) случайный лес (Random Forest);
 - c) кластеризация K-means;
 - d) градиентный бустинг.
14. Какова главная цель применения метода главных компонент (PCA)?
- a) увеличение количества признаков для повышения точности;
 - b) снижение размерности данных при сохранении максимальной дисперсии (информации);
 - c) классификация текстовых документов по тональности;
 - d) генерация синтетических тестовых данных.
15. Какую роль выполняют брокеры сообщений (например, RabbitMQ или Apache Kafka) в микросервисной архитектуре ИИ-системы?
- a) хранение обученных весов нейронных сетей;
 - b) обеспечение асинхронного обмена сообщениями и интеграция между микросервисами;
 - c) визуализация метрик качества модели;
 - d) непосредственное обучение моделей на GPU.
16. Что такое трансферное обучение (Transfer Learning)?
- a) перенос вычислений с центрального процессора (CPU) на графический ускоритель (GPU);
 - b) использование предварительно обученной модели, дообученной для решения новой, но схожей задачи;
 - c) трансляция программного кода с языка Python на язык C++;
 - d) процесс миграции базы данных из локального сервера в облако.
17. Что означает термин «Ложноположительное срабатывание» (False Positive) в матрице ошибок (Confusion Matrix)?
- a) модель верно предсказала положительный класс;
 - b) модель ошибочно предсказала положительный класс, хотя истинное значение было отрицательным;
 - c) модель верно предсказала отрицательный класс;

- d) модель ошибочно предсказала отрицательный класс, хотя истинное значение было положительным.
18. Какова функция оптимизатора (например, SGD, Adam) в процессе обучения нейронной сети?
- a) инициализация весов случайными числами перед началом обучения;
 - b) обновление весов сети на основе градиентов для минимизации функции потерь;
 - c) применение функции активации к выходам нейронов;
 - d) объединение признаков в пулинговых слоях.
19. Чем гиперпараметры модели машинного обучения отличаются от её параметров (весов)?
- a) гиперпараметры вычисляются автоматически в процессе обучения, а параметры задает человек;
 - b) гиперпараметры задаются до начала обучения (например, скорость обучения, количество деревьев), а параметры модель находит сама в данных;
 - c) гиперпараметры используются только в нейронных сетях, а параметры — только в линейной регрессии;
 - d) между ними нет разницы, это синонимы.
20. Для чего проводится A/B тестирование ИИ-моделей в промышленной эксплуатации?
- a) для ускорения сходимости градиентного спуска;
 - b) для сравнения эффективности двух версий модели (или функций) на реальных пользователях с целью выбора лучшей;
 - c) для сжатия (квантизации) модели перед развертыванием на мобильных устройствах;
 - d) для автоматической очистки данных от выбросов.

МДК 01.02. Разработка мобильных приложений с поддержкой
искусственного интеллекта

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК
1.5)

1. Какая система сборки используется по умолчанию в среде Android Studio для управления зависимостями, конфигурациями и процессом компиляции проекта?
- a) Maven;
 - b) Gradle;
 - c) CMake;
 - d) Ant.

2. Какой компонент пользовательского интерфейса в Android рекомендуется использовать для эффективного отображения больших списков или сеток данных с поддержкой прокрутки и переиспользования элементов?
 - a) ListView;
 - b) ScrollView;
 - c) RecyclerView;
 - d) LinearLayout.
3. Какой фреймворк от Google предназначен специально для запуска и оптимизации моделей машинного обучения непосредственно на мобильных и встроенных устройствах?
 - a) PyTorch;
 - b) TensorFlow Lite;
 - c) Scikit-learn;
 - d) Keras.
4. Какая современная библиотека Android (Jetpack) используется для упрощенного доступа к камере устройства и передачи видеопотока в ИИ-модель в реальном времени?
 - a) Camera2;
 - b) CameraX;
 - c) OpenCV;
 - d) MediaCodec.
5. Какой метод оптимизации ИИ-моделей для мобильных устройств заключается в снижении битности числовых представлений весов (например, с float32 до int8) для уменьшения размера модели и ускорения инференса?
 - a) Прунинг (Pruning / Отсечение);
 - b) Квантизация (Quantization);
 - c) Аугментация (Augmentation);
 - d) Дистилляция (Distillation).
6. Какой программный интерфейс Android позволяет переносить вычисления нейронных сетей на аппаратные ускорители устройства (GPU, DSP, NPU) для повышения производительности?
 - a) OpenGL ES;
 - b) Vulkan;
 - c) NNAPI (Neural Networks API);
 - d) RenderScript.
7. Как называется архитектура мобильного ИИ-приложения, при которой базовые функции работают локально без доступа к интернету, а сложные задачи при наличии сети отправляются на сервер?

- a) Cloud-native;
 - b) Гибридная архитектура (Offline-first);
 - c) Serverless;
 - d) Микросервисная архитектура.
8. Какой формат файла рекомендуется использовать для публикации приложений в Google Play Console, чтобы обеспечить автоматическую оптимизацию размера приложения под конфигурации разных устройств?
- a) APK (Android Package);
 - b) AAB (Android App Bundle);
 - c) IPA (iOS App Store Package);
 - d) XAP.
9. Какой инструмент экосистемы Firebase чаще всего используется для мониторинга сбоев (crashes) и ошибок ANR (Application Not Responding) в мобильных приложениях после релиза?
- a) Google Analytics;
 - b) Firebase Crashlytics;
 - c) Firebase Remote Config;
 - d) Firebase Cloud Messaging.
10. Где в Android-приложении рекомендуется безопасно хранить конфиденциальные данные, такие как криптографические ключи или токены аутентификации?
- a) В файле strings.xml;
 - b) В открытом виде в коде на Kotlin/Java;
 - c) В Android Keystore или зашифрованных SharedPreferences;
 - d) В файле build.gradle.
11. Какая популярная модель ветвления в Git предполагает использование веток main, develop, feature, release и hotfix для организации командной разработки и релизного цикла?
- a) Trunk-based development;
 - b) GitFlow;
 - c) GitHub Flow;
 - d) GitLab Flow.
12. Какая технология позволяет накладывать виртуальные 3D-объекты или информационные слои на изображение с камеры в реальном времени?
- a) VR (Виртуальная реальность);
 - b) AR (Дополненная реальность);
 - c) MR (Смешанная реальность);
 - d) XR (Расширенная реальность).
13. Какой вид тестирования мобильного приложения предполагает сравнение

- двух версий интерфейса или функции на реальных пользователях для выявления более эффективной с точки зрения бизнес-метрик?
- a) Модульное тестирование (Unit testing);
 - b) Интеграционное тестирование;
 - c) A/B тестирование;
 - d) Стресс-тестирование.
14. Что из перечисленного является специфической угрозой безопасности именно для ИИ-моделей, встроенных в мобильные приложения?
- a) SQL-инъекция в локальную базу данных;
 - b) Кража или реверс-инжиниринг весов модели из APK/AAB файла;
 - c) XSS-атака на веб-интерфейс;
 - d) DDoS-атака на серверную часть.
15. Какая задача в области обработки естественного языка (NLP) решается при создании функции голосового ввода или диктовки в мобильном приложении?
- a) OCR (Оптическое распознавание символов);
 - b) STT (Speech-to-Text / Распознавание речи);
 - c) Object Detection (Обнаружение объектов);
 - d) Image Segmentation (Сегментация изображений).
16. Какой инструмент в Android Studio позволяет визуально анализировать иерархию представлений (View Hierarchy) и находить узкие места в производительности отрисовки UI?
- a) Logcat;
 - b) Layout Inspector;
 - c) Memory Profiler;
 - d) Network Inspector.
17. Как называется процесс удаления избыточных связей или нейронов в нейронной сети без значительной потери точности для уменьшения вычислительной сложности модели?
- a) Квантизация;
 - b) Прунинг (Pruning);
 - c) Аугментация;
 - d) Нормализация.

МДК 01.03. Тестирование программных модулей
(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.6, ПК 1.7)

1. Какой документ определяет общую стратегию, цели, ресурсы и расписание тестирования проекта?
- a) тестовый сценарий (Test Case);

- b) тестовый план (Test Plan);
 - c) чек-лист (Checklist);
 - d) отчет о дефекте (Defect Report).
2. Какой вид тестирования направлен на проверку взаимодействия между отдельными модулями или компонентами системы?
- a) модульное тестирование (Unit testing);
 - b) интеграционное тестирование (Integration testing);
 - c) приемочное тестирование (Acceptance testing);
 - d) регрессионное тестирование (Regression testing).
3. Какая метрика качества модели классификации вычисляется как отношение верно предсказанных положительных классов к общему количеству реально положительных объектов?
- a) precision (точность);
 - b) recall (полнота);
 - c) accuracy (аккуратность);
 - d) F1-score.
4. Какой метод тест-дизайна основан на разделении входных данных на классы, внутри которых программа ведет себя одинаково?
- a) анализ граничных значений;
 - b) эквивалентное разбиение (equivalence partitioning);
 - c) таблицы решений;
 - d) тестирование переходов состояний;
5. Что такое AUC (Area Under Curve) в контексте оценки качества модели машинного обучения?
- a) средняя абсолютная ошибка прогноза;
 - b) площадь под ROC-кривой, отражающая способность модели различать классы;
 - c) процент правильно классифицированных объектов;
 - d) коэффициент детерминации модели;
6. Какой тип тестирования проверяет, как система ведет себя при превышении расчетной нагрузки?
- a) функциональное тестирование;
 - b) стресс-тестирование (stress testing);
 - c) дымовое тестирование (smoke testing);
 - d) санитарное тестирование (sanity testing).
7. Какая техника тест-дизайна используется для проверки сложной бизнес-логики с множеством комбинаций условий?
- a) метод эквивалентного разбиения;
 - b) таблицы решений (decision tables);

- c) анализ граничных значений;
 - d) попарное тестирование (pairwise testing).
8. Что такое E2E-тестирование (End-to-End testing)?
- a) тестирование отдельной функции изолированно от других модулей;
 - b) тестирование полного пользовательского сценария от начала до конца, имитирующее реальное использование системы;
 - c) тестирование только интерфейса пользователя без проверки бизнес-логики;
 - d) тестирование производительности базы данных.
9. Какая метрика используется для оценки качества моделей регрессии и показывает среднюю абсолютную разницу между предсказанными и реальными значениями?
- a) F1-score;
 - b) MAE (Mean Absolute Error);
 - c) AUC-ROC;
 - d) Confusion Matrix.
10. Какой инструмент преимущественно используется для автоматизации тестирования API ИИ-сервисов?
- a) Selenium;
 - b) Postman / SoapUI;
 - c) JUnit;
 - d) TestComplete.
11. Что такое «дрейф данных» (Data Drift) в контексте мониторинга ИИ-систем?
- a) процесс миграции данных из локального хранилища в облако;
 - b) изменение статистического распределения входных данных со временем, приводящее к деградации качества модели;
 - c) ошибка при передаче данных по сети;
 - d) метод сжатия данных для экономии памяти.
12. Какой вид тестирования направлен на повторную проверку ранее исправленных дефектов и оценку того, что новые изменения не сломали существующий функционал?
- a) модульное тестирование;
 - b) регрессионное тестирование;
 - c) альфа-тестирование;
 - d) A/B тестирование.
13. Что такое мок (mock) в контексте автоматизированного тестирования ИИ-систем?
- a) реальный API-сервис для интеграционного тестирования;

- b) имитация внешнего объекта или сервиса с заранее заданным поведением для изоляции тестируемого модуля;
 - c) ошибка в тестовом сценарии;
 - d) инструмент для генерации тестовых данных.
- 14.Какая метрика классификации представляет собой гармоническое среднее между Precision и Recall?
- a) Accuracy;
 - b) ROC-AUC;
 - c) F1-score;
 - d) MAPE.
- 15.Какой стандарт описывает характеристики качества программного обеспечения (функциональная пригодность, производительность, совместимость, удобство использования, надежность, безопасность)?
- a) ISO 9001;
 - b) ГОСТ Р ИСО/МЭК 25051;
 - c) ITIL v4;
 - d) PMBOK.
- 16.Что такое покрытие кода (code coverage) в тестировании?
- a) процент требований, закрытых тестовыми сценариями;
 - b) метрика, показывающая, какая доля строк, ветвей или условий исходного кода была выполнена во время тестирования;
 - c) количество тестовых сценариев в тестовом пакете;
 - d) объем документации, приложенной к тестовому плану.
- 17.Какой вид тестирования ИИ-системы предполагает сравнение двух версий модели на реальных пользователях для выбора более эффективной?
- a) модульное тестирование;
 - b) A/B тестирование;
 - c) статическое тестирование;
 - d) санитарное тестирование;
- 18.Какая техника тест-дизайна наиболее эффективна для проверки поведения системы на крайних значениях входного диапазона (например, 0, -1, максимальное значение)?
- a) таблицы решений;
 - b) анализ граничных значений (boundary value analysis);
 - c) тестирование переходов состояний;
 - d) use case testing.
- 19.Что такое матрица ошибок (Confusion Matrix)?
- a) таблица с описанием всех дефектов, найденных в ходе тестирования;
 - b) таблица 2x2 (или NxN), показывающая соответствие между реальными

- и предсказанными классами модели (TP, TN, FP, FN);
- с) документ с результатами нагрузочного тестирования;
- д) схема архитектуры ИИ-системы.
20. Какой подход к тестированию ИИ-моделей предполагает проверку устойчивости модели к специально сконструированным входным данным, вводящим модель в заблуждение?
21. юзабилити-тестирование;
22. adversarial testing (сопоставительное тестирование);
23. локализация;
24. кросс-платформенное тестирование.

Критерии оценки знаний студентов при проведении тестирования

Оценка «отлично» выставляется при условии правильного ответа студента не менее чем на 86 % тестовых заданий;

Оценка «хорошо» выставляется при условии правильного ответа студента на 70-85 % тестовых заданий;

Оценка «удовлетворительно» выставляется при условии правильного ответа студента на 50-69 % тестовых заданий;

Оценки «неудовлетворительно» заслуживает студент при условии правильного ответа студента менее чем на 50 % тестовых заданий.

Практические задания

МДК 01.01. Разработка программных модулей в системах искусственного интеллекта

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

Задание 1. Провести сравнительный анализ различных направлений ИИ (машинное обучение, глубокое обучение, нейронные сети) на примере решения прикладной задачи. Реализовать прототип базовой модели ИИ для классификации данных (или базовой системы рекомендаций) с использованием стандартных библиотек, проанализировать результаты и оформить отчет о проделанной работе.

Задание 2. Осуществить сбор данных из открытого источника (с использованием веб-скрапинга или API). Выполнить первичный анализ и визуализацию датасета. Провести очистку данных от пропусков и выбросов, нормализацию и стандартизацию. Применить методы Feature Engineering для создания новых признаков и подготовить несбалансированные данные для дальнейшего обучения модели.

Задание 3. Реализовать на заданном наборе данных алгоритмы

машинного обучения с учителем (например, метод k-ближайших соседей, метод опорных векторов или случайный лес) и без учителя (кластеризация k-means или метод главных компонент PCA). Провести сравнительный анализ эффективности выбранных алгоритмов, построить систему рекомендаций или выявить скрытые паттерны в данных с помощью ассоциативных правил (Apriori).

Задание 4. Разделить исходный датасет на обучающую и тестовую выборки. Провести кросс-валидацию (включая стратифицированную и валидацию на временных рядах). Оценить качество обученной модели путем построения матрицы ошибок (Confusion Matrix), ROC-кривой и кривых Precision-Recall. Выполнить настройку гиперпараметров с использованием GridSearchCV и применить L1/L2-регуляризацию для предотвращения переобучения и улучшения алгоритма.

Задание 5. Спроектировать и визуализировать архитектуру нейронной сети (сверточной CNN для задачи распознавания изображений или рекуррентной LSTM для анализа временных рядов). Исследовать влияние различных функций активации (ReLU, сигмоида) и оптимизаторов на процесс обучения. Применить методы аугментации данных, пакетную нормализацию и трансферное обучение (Transfer learning) с использованием предобученных сетей для повышения точности модели. Сохранить обученную модель для последующего развертывания.

Задание 6. Спроектировать микросервисную архитектуру для ИИ-системы. Выполнить контейнеризацию разработанной ИИ-модели с использованием Docker и настроить ее развертывание. Создать REST API для интеграции ИИ-сервиса с внешними системами. Настроить процесс версионирования моделей и данных (с использованием MLflow или DVC), а также реализовать базовый CI/CD пайплайн для автоматического тестирования и обновления модели. Провести нагрузочное тестирование системы и настроить мониторинг ее надежности.

МДК 01.02. Разработка мобильных приложений с поддержкой искусственного интеллекта

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

Задание 1: Создание базового каркаса мобильного приложения. Настройка среды Android Studio и конфигурации Gradle. Разработка многоэкранного пользовательского интерфейса с использованием Navigation Component, RecyclerView и ConstraintLayout. Реализация обработки событий взаимодействия с пользователем, внедрение темной темы и поддержка базовых требований доступности (Accessibility).

Результат выполнения: Работающий прототип Android-приложения с настроенной навигацией между экранами, адаптивной версткой и корректной обработкой UI-событий.

Задание 2. Интеграция предобученной TensorFlow Lite модели в Android-приложение. Подключение камеры через библиотеку CameraX для захвата видеопотока и передачи кадров в ИИ-модель в реальном времени (например, для классификации изображений или обнаружения объектов). Применение квантизации модели для уменьшения ее размера и настройка аппаратного ускорения инференса (GPU Delegate).

Результат выполнения: Мобильное приложение, способное в реальном времени анализировать изображение с камеры устройства с использованием оптимизированной локальной ИИ-модели.

Задание 3. Разработка интерактивного модуля приложения с использованием встроенных сенсоров и ИИ. Создание голосового помощника или модуля распознавания жестов с реализацией мгновенной визуальной и звуковой обратной связи для пользователя. Настройка обработки естественного языка (NLP) или интеграция AR-компонентов для наложения виртуальных объектов на реальный мир.

Результат выполнения: Интерактивный модуль приложения, использующий микрофон, камеру или акселерометр в связке с ИИ-алгоритмами для обеспечения бесшовного взаимодействия с пользователем.

Задание 4: Подготовка мобильного ИИ-приложения к релизу. Настройка CI/CD пайплайна для автоматического запуска тестов. Оптимизация размера итоговой сборки (формат AAB), выполнение криптографической подписи приложения и настройка мониторинга сбоев (Crashlytics) и ошибок ANR после выпуска версии в Google Play Console.

Результат выполнения: Подписанная и оптимизированная сборка приложения, готовая к публикации, с настроенным пайплайном непрерывной интеграции и подключенными системами пост-релизной аналитики.

Задание 5: Реализация подсистемы безопасности мобильного приложения. Настройка биометрической аутентификации и шифрование локальных данных. Обеспечение безопасного хранения API-ключей и защита сетевого трафика. Внедрение методов обфускации кода и защиты файла ИИ-модели от несанкционированного извлечения и реверс-инжиниринга.

Результат выполнения: Защищенный модуль приложения с реализованной аутентификацией, шифрованием данных и мерами по предотвращению кражи интеллектуальной собственности (ИИ-модели).

Задание 6: Организация командной работы и сопровождения проекта. Настройка ветвления по модели GitFlow, проведение Code Review и

написание технической документации. Интеграция систем аналитики для сбора обратной связи, настройка А/В тестирования новой ИИ-функции и подготовка локализованной версии приложения для выпуска планового обновления.

Результат выполнения: Настроенный процесс командной разработки, документация проекта, подключенная аналитика и настроенный механизм А/В тестирования для оценки эффективности новых функций.

МДК 01.03. Тестирование программных модулей (ОК 01; ОК 02, ОК 04, ОК 05, ОК 09, ПК 1.6, ПК 1.7)

Задание 1: Разработка модели управления дефектами и построение диаграммы причинно-следственных связей. Студенту необходимо классифицировать виды тестирования для заданного ИИ-проекта, описать жизненный цикл дефекта с учетом специфики ИИ (например, учет дрейфа данных), а также построить диаграмму причинно-следственных связей для сложного бизнес-правила системы и вывести на ее основе первичные тестовые условия.

Результат выполнения: Документ с классификацией тестов, регламент управления дефектами (с матрицей серьезности и приоритета) и графическая диаграмма причинно-следственных связей с вытекающими тестовыми условиями.

Задание 2: Разработка тестовой стратегии и пакета тестовых артефактов для ИИ-модуля. Студент должен создать тестовый план, разработать чек-листы для смоук-тестирования ИИ-сервиса и подготовить детальные тестовые сценарии (тест-кейсы) с использованием стандартных шаблонов. В конце необходимо составить матрицу трассировки для оценки покрытия тестами бизнес-требований.

Результат выполнения: Комплект документации: Тестовый план, чек-листы, пакет тестовых сценариев и матрица трассировки требований.

Задание 3: Проектирование тестов с использованием спецификационных методов «черного ящика». Студенту необходимо применить метод эквивалентного разбиения и анализа граничных значений для входных параметров ИИ-модели, составить таблицу решений для сложной логики классификатора (с множеством условий) и разработать сценарии тестирования на основе диаграмм Use Case.

Результат выполнения: Набор тестовых сценариев, разработанных с применением техник эквивалентного разбиения, таблиц решений и Use Case, с расчетом процента покрытия бизнес-логики.

Задание 4: Оценка качества модели машинного обучения с

использованием метрик классификации и регрессии. Студенту предоставляются массивы предсказаний модели и фактические значения (ground truth). Необходимо построить матрицу ошибок (Confusion Matrix), рассчитать Accuracy, Precision, Recall, F1-score, а также метрики регрессии (MAE, RMSE, MAPE). На основе расчетов сделать вывод о пригодности модели для бизнес-задачи с учетом стоимости ложноположительных и ложноотрицательных срабатываний.

Результат выполнения: Отчет об оценке качества модели с расчетами метрик, визуализацией матрицы ошибок и обоснованием выбора оптимального порога отсечения (threshold).

Задание 5: Разработка автоматизированных тестов и настройка CI/CD пайплайна для ИИ-модуля. Студент должен написать юнит-тесты для функций предобработки данных, создать моки (mocks) и стабы (stubs) для изоляции внешних API-зависимостей ИИ-сервиса, а также настроить скрипт автоматизации регрессионного тестирования с генерацией отчета в рамках CI/CD пайплайна.

Результат выполнения: Исходный код автоматизированных тестов (с использованием mock-объектов) и конфигурационный файл CI/CD пайплайна с шагом запуска тестов и генерации отчета.

Задание 6: Настройка тестового окружения и тестирование API ИИ-сервиса. Студент должен подготовить тестовый стенд с использованием заглушек для имитации базы данных и внешних сервисов, разработать коллекцию запросов в Postman/SoapUI для тестирования REST API ИИ-сервиса (проверка валидации входных данных, обработки исключений, проверки формата JSON и времени отклика) и провести комплексное приемочное тестирование перед релизом.

Результат выполнения: Коллекция API-тестов (Postman/SoapUI), описание настроенного тестового окружения со заглушками и отчет о результатах приемочного тестирования API.

Задания для лабораторных работ

МДК 01.01. Разработка программных модулей в системах искусственного интеллекта

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

Задание 1. Провести эксперимент по применению готовых предобученных моделей (например, из библиотеки Transformers или OpenCV) для задач классификации изображений и анализа тональности текста. Построить интерактивный интерфейс (с использованием Streamlit или Gradio)

для подачи входных данных, визуализации промежуточных результатов работы модели и анализа степени уверенности модели в своих предсказаниях.

Задание 2. Выполнить сравнительный анализ различных стратегий обработки пропущенных значений (удаление, заполнение средним/медианой, интерполяция, предсказание с помощью KNN) и методов масштабирования (StandardScaler, MinMaxScaler, RobustScaler). Построить гистограммы распределений, ящики с усами (boxplots) и корреляционные матрицы (heatmaps) до и после применения методов. Сделать количественный вывод о влиянии каждого метода на структуру и дисперсию исходного датасета.

Задание 3. Сгенерировать двумерные синтетические наборы данных с различными типами разделимости (линейная, нелинейная, с пересечением классов). Обучить на них алгоритмы (kNN, SVM с различными ядрами, логистическую регрессию). Построить графики границ принятия решений (decision boundaries) и тепловые карты вероятностей предсказания классов. Проанализировать, как изменение гиперпараметров (например, параметра C в SVM или количества соседей в kNN) влияет на форму и сложность границы.

Задание 4. Провести серию экспериментов по обучению моделей с контролируемой сложностью (например, полиномиальная регрессия разной степени или деревья решений с ограничением глубины) на зашумленных данных. Построить кривые обучения (learning curves) и кривые валидации (validation curves). Рассчитать количественные значения смещения (bias), дисперсии (variance) и неустраняемой ошибки для каждой конфигурации. Определить оптимальную сложность модели, минимизирующую ошибку на тестовой выборке.

Задание 5. Спроектировать идентичную архитектуру многослойного перцептрона (MLP) и провести серию запусков, варьируя только функции активации (ReLU, Sigmoid, Tanh, Leaky ReLU) и алгоритмы оптимизации (SGD, Momentum, Adam, RMSprop). С помощью инструментов визуализации (например, TensorBoard) отследить динамику изменения функции потерь (loss) и точности (accuracy) на каждой эпохе. Проанализировать скорость сходимости, наличие проблем затухающего/взрывающегося градиента и итоговое качество моделей.

Задание 6. Установить и сконфигурировать локальный сервер системы управления жизненным циклом машинного обучения (например, MLflow Tracking Server). Модифицировать скрипт обучения модели таким образом, чтобы при каждом запуске автоматически логировались гиперпараметры, метрики качества, версии библиотек и сохранялись артефакты (файлы модели). Настроить дашборд для визуального сравнения различных запусков (runs) и анализа влияния гиперпараметров на целевую метрику.

МДК 01.02. Разработка мобильных приложений с поддержкой
искусственного интеллекта

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

Задание 1: Провести эксперимент по сравнению производительности различных UI-компонентов (например, ListView vs RecyclerView, LinearLayout vs ConstraintLayout) при отображении больших объемов данных. Использовать инструменты профилирования (Android Profiler, Layout Inspector) для измерения времени отрисовки (render time), использования памяти и загрузки процессора. Построить графики зависимости времени рендеринга от количества элементов в списке. Сделать выводы о целесообразности использования тех или иных компонентов в зависимости от конкретных задач и ограничений устройства.

Результат выполнения: Отчет со сравнительными графиками производительности, скриншотами из профилировщика и обоснованными рекомендациями по выбору UI-компонентов.

Задание 2: Взять одну и ту же базовую модель машинного обучения (например, для классификации изображений) и подготовить её в нескольких вариантах: без оптимизации (float32), с применением квантизации (INT8) и с применением прунинга. Интегрировать каждый вариант в тестовое Android-приложение. Провести серию экспериментов по измерению времени инференса (латентности), размера итогового файла приложения, потребления оперативной памяти и заряда батареи. Оценить компромисс между точностью модели (построить Confusion Matrix для каждого варианта) и её производительностью на мобильном устройстве.

Результат выполнения: Сводная таблица и графики, демонстрирующие trade-off (компромисс) между точностью, размером модели, скоростью работы и потреблением энергии.

Задание 3: Разработать тестовый стенд для сбора данных с различных сенсоров устройства (акселерометр, гироскоп, микрофон, камера) с настраиваемой частотой дискретизации. Подать собранные данные на вход локальной ИИ-модели (например, для распознавания жестов или голосовых команд). Исследовать влияние задержек (latency) сенсоров, уровня шума и частоты опроса на точность предсказаний модели. Построить графики зависимости метрик качества (Accuracy, F1-score) от частоты обновления данных сенсоров.

Результат выполнения: Аналитический отчет с графиками и выработанными рекомендациями по оптимальным настройкам сенсоров и частоте опроса для обеспечения бесшовного интерактивного взаимодействия.

Задание 4: Настроить эмулятор мобильного устройства с различными профилями сети (3G, 4G, Wi-Fi, Lossy network с потерей пакетов, Offline). Запустить приложение с гибридной архитектурой ИИ (локальный инференс + облачный API). С помощью инструментов мониторинга (Firebase Performance Monitoring, Android Profiler) собрать телеметрию: время отклика, процент успешных запросов, частота сбоев (Crash rate), потребление трафика. Провести стресс-тестирование при резком переключении между сетями и уходе в офлайн.

Результат выполнения: Дашборд с ключевыми метриками стабильности и производительности, анализ поведения приложения в пограничных сетевых условиях и список выявленных узких мест.

Задание 5: Провести статический (SAST) и динамический (DAST) анализ безопасности тестового мобильного приложения, содержащего ИИ-модель. Использовать специализированные инструменты (например, MobSF, Drozer, Burp Suite) для поиска уязвимостей: небезопасное хранение данных, утечки через логи, уязвимости сетевого трафика, возможность извлечения весов ИИ-модели из APK/AAB-файла. Попытаться провести атаку на модель (Model Extraction Attack) через API. Составить отчет о найденных уязвимостях с оценкой их критичности (CVSS) и предложить конкретные методы их устранения.

Результат выполнения: Отчет о тестировании на проникновение (Pentest Report) с классификацией уязвимостей и планом мероприятий по их устранению (шифрование, обфускация, certificate pinning).

Задание 6: Настроить систему продуктовой аналитики для тестового мобильного приложения. Спроектировать и провести виртуальный A/B тест новой ИИ-функции (например, умные рекомендации или автоматическое улучшение фото). Используя открытый датасет пользовательских событий (клики, время в приложении, конверсия) или сгенерировав синтетические данные, провести статистический анализ результатов A/B теста. Рассчитать p-value, доверительные интервалы и статистическую мощность теста.

Результат выполнения: Отчет о результатах A/B тестирования с математическим обоснованием статистической значимости различий между контрольной и тестовой группами и финальным бизнес-решением о раскатке функции.

МДК 01.03.Тестирование программных модулей
(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.6, ПК 1.7)

Задание 1: Студенту предоставляется набор данных о дефектах, найденных в ходе тестирования ИИ-системы, и статистика времени отклика

модулей. Необходимо рассчитать ключевые метрики качества (функциональная пригодность, производительность, надежность, удобство использования) в соответствии со стандартом. Построить графики распределения дефектов по серьезности и приоритету. Проанализировать жизненный цикл дефекта и оценить эффективность процесса управления дефектами.

Результат выполнения: Отчет с расчетами метрик качества, графиками распределения дефектов и выводами о соответствии системы требованиям ГОСТ.

Задание 2: Студент должен разработать универсальные шаблоны тестовых сценариев (test case templates) для различных типов тестирования (функциональное, интеграционное, регрессионное). На основе предоставленного списка бизнес-требований к ИИ-модулю составить матрицу трассировки (traceability matrix) и рассчитать процент покрытия требований тестами. Проанализировать gaps (пробелы) в покрытии и предложить дополнительные тестовые сценарии.

Результат выполнения: Набор шаблонов тестов, матрица трассировки требований с расчетом процента покрытия и рекомендации по устранению пробелов.

Задание 3: Студенту предоставляется фрагмент исходного кода ИИ-модуля (например, функция предобработки данных или классификации). Необходимо применить различные техники тест-дизайна (эквивалентное разбиение, анализ граничных значений, таблицы решений, тестирование переходов состояний) для генерации тестовых сценариев. Выполнить рефакторинг тестовых сценариев для оптимизации их количества. С помощью инструментов анализа покрытия кода (code coverage tools) рассчитать процент покрытия строк и ветвей.

Результат выполнения: Набор оптимизированных тестовых сценариев, отчет о покрытии кода с визуализацией и анализ эффективности различных техник тест-дизайна.

Задание 4: Студенту предоставляется набор данных и серия моделей машинного обучения с различной сложностью (например, полиномиальная регрессия разной степени или деревья решений с ограничением глубины). Необходимо построить кривые обучения (learning curves) и кривые валидации (validation curves) для каждой модели. Рассчитать количественные значения смещения (bias), дисперсии (variance) и общей ошибки. Определить оптимальную сложность модели, минимизирующую ошибку на тестовой выборке, и обосновать выбор.

Результат выполнения: Графики кривых обучения и валидации,

таблица с расчетами bias-variance tradeoff и обоснование выбора оптимальной модели.

Задание 5: Студент должен разработать параметризованные автоматизированные тесты для ИИ-модели, которые позволяют тестировать различные комбинации входных параметров и гиперпараметров модели. Использовать генераторы синтетических тестовых данных для создания граничных случаев и edge cases. Провести массовый запуск параметризованных тестов и проанализировать результаты. Выявить закономерности в поведении модели при различных входных данных.

Результат выполнения: Исходный код параметризованных тестов, набор сгенерированных синтетических данных и отчет с анализом результатов массового тестирования.

Задание 6: Студенту предоставляются данные о работе двух версий ИИ-модели (контрольная группа А и тестовая группа В) в production-среде. Необходимо провести статистический анализ результатов А/В теста: рассчитать конверсию, среднее время отклика, процент ошибок. Определить статистическую значимость различий между группами (рассчитать p-value, доверительные интервалы). Сделать вывод о том, какая версия модели показывает лучшие бизнес-показатели, и обосновать рекомендацию по раскатке.

Результат выполнения: Отчет о результатах А/В тестирования с расчетами статистической значимости, графиками распределения метрик и финальной рекомендацией для бизнес-заказчика.

Критерии оценки выполнения практических заданий (заданий для лабораторных работ)

Оценка «отлично» выставляется студенту, если практическое задание (задание для лабораторных работ) выполнено в полном объеме с соблюдением необходимой последовательности действий; итогом решения задания явилось получение правильных результатов и выводов; в ходе решения задания верно и аккуратно выполнены все вычисления и записи.

Оценка «хорошо» выставляется студенту, если практическое задание (задание для лабораторных работ) выполнено правильно с учетом мелких погрешностей или недочетов.

Оценка «удовлетворительно» выставляется студенту, если практическое задание (задание для лабораторных работ) выполнено правильно не менее чем наполовину, допущены некоторые погрешности или одна грубая ошибка.

Оценки «неудовлетворительно» заслуживает студент, если при решении практического задания (задания для лабораторных работ) допущены две (и более) грубые ошибки, или работа не выполнена полностью.

Кейс-задания

МДК 01.01. Разработка программных модулей в системах искусственного интеллекта
(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

Кейс-задание 1.

Кейс-сценарий: Крупная розничная сеть обратилась к вашей компании с запросом на внедрение технологий ИИ. У компании есть три проблемы: 1) падение продаж из-за нерелевантных предложений клиентам; 2) высокая нагрузка на колл-центр из-за типовых вопросов; 3) очереди на кассах в часы пик, которые приводят к оттоку покупателей. В распоряжении компании есть база данных покупок, логи чатов и звонков, а также видеокамеры в торговых залах.

Задание: Проанализируйте бизнес-задачи клиента. Предложите по одному решению на базе ИИ для каждой из трех проблем. Для каждого решения обоснуйте выбор направления ИИ (машинное обучение, обработка естественного языка, компьютерное зрение), опишите, какие именно данные потребуются для обучения моделей, и выявите потенциальные этические риски и риски приватности при внедрении данных систем.

Результат выполнения: Аналитическая записка с матрицей «Проблема — Решение — Направление ИИ — Требуемые данные — Риски».

Кейс-задание 2.

Кейс-сценарий: Медицинский стартап разрабатывает систему прогнозирования рисков сердечно-сосудистых заболеваний. Данные поступают из трех разных клиник в разных форматах: CSV-таблицы с показателями анализов (где 30% значений пропущено), неструктурированные текстовые заметки врачей и изображения ЭКГ. Кроме того, заболевание встречается лишь у 5% пациентов в выборке. Юридический отдел требует соблюдения строгих норм обезличивания данных.

Задание: Вы выступаете в роли Lead Data Engineer. Спроектируйте конвейер (pipeline) подготовки данных. Опишите стратегию сбора и безопасного хранения данных. Предложите конкретные методы обработки пропусков для числовых и текстовых данных. Разработайте план действий по работе с сильным дисбалансом классов (5% против 95%) и предложите 3 идеи для Feature Engineering на основе медицинских показателей.

Результат выполнения: Схема конвейера данных (Data Pipeline) и

техническое обоснование выбранных методов предобработки.

Кейс-задание 3.

Кейс-сценарий: Банк поручает вашей команде разработать модель скоринга для одобрения кредитов. Руководство банка требует, чтобы модель была максимально точной. Однако юридический департамент выдвигает жесткое условие: модель должна быть интерпретируемой. При отказе в кредите менеджер должен иметь возможность объяснить клиенту конкретные причины (например, «отказ из-за высокой долговой нагрузки и отсутствия кредитной истории»), а не просто сослаться на «решение черного ящика».

Задание: Проведите сравнительный анализ трех алгоритмов: Логистическая регрессия, Случайный лес (Random Forest) и Градиентный бустинг (XGBoost/LightGBM) в контексте данного ТЗ. Выберите оптимальный алгоритм, обоснуйте свой выбор с точки зрения баланса между точностью и интерпретируемостью. Опишите, как вы будете кодировать категориальные признаки (тип занятости, семейное положение) для выбранной модели.

Результат выполнения: Сравнительная таблица алгоритмов и обоснованный выбор финального решения для защиты перед заказчиком.

Кейс-задание 4.

Кейс-сценарий: Телеком-компания разработала модель для прогнозирования оттока клиентов (Churn Rate). На тестовой выборке модель показала Ассигасу (точность) 94%. Руководство было радо, но после внедрения маркетинговый отдел сообщил, что модель бесполезна: она почти не находит тех, кто реально уходит, и тратит бюджет на удержание лояльных клиентов. В выборке 94% клиентов остаются и 6% уходят.

Задание: Объясните руководству, почему метрика Ассигаса ввела их в заблуждение (парадокс точности). Определите, какие метрики (Precision, Recall, F1-score, ROC-AUC) необходимо использовать в данном случае, и обоснуйте свой выбор с точки зрения бизнес-логики (что для компании дороже: ложноположительное или ложноотрицательное срабатывание?). Предложите 3 конкретных метода (на уровне данных или алгоритма) для улучшения способности модели находить «уходящих» клиентов.

Результат выполнения: Отчет об анализе ошибок модели с расчетом бизнес-стоимости ошибок и планом по улучшению метрики Recall.

Кейс-задание 5.

Кейс-сценарий: Агротехническая компания внедряет дрон для мониторинга состояния посевов. Дрон должен в реальном времени распознавать болезни растений по фотографиям листьев. Проблема: текущая сверточная нейронная сеть (CNN) показывает высокую точность на сервере, но она слишком «тяжелая» — время инференса составляет 2 секунды, а

батарея дрона садится за 10 минут. При этом размеченных фотографий болезней растений у компании всего 5 000 штук.

Задание: Вам необходимо оптимизировать решение для работы на Edge-устройстве (бортовом компьютере дрона). Предложите комплексную стратегию решения проблемы. Во-первых, как повысить качество модели при малом объеме данных (аугментация, трансферное обучение)? Во-вторых, какие методы оптимизации и сжатия модели (квантизация, прунинг, выбор легкой архитектуры) вы примените для ускорения инференса и снижения потребления ресурсов без критической потери точности?

Результат выполнения: Техническое задание на оптимизацию нейросети с описанием выбранных архитектурных решений и методов сжатия.

Кейс-задание 6.

Кейс-сценарий: Финтех-компания запускает систему обнаружения мошеннических транзакций (Fraud Detection) в реальном времени. Система должна обрабатывать до 10 000 транзакций в секунду с задержкой (latency) не более 50 мс. Кроме того, команда Data Science хочет иметь возможность обновлять модели машинного обучения каждую неделю без остановки сервиса (Zero Downtime), а в случае падения качества новой модели — мгновенно откатываться на предыдущую версию.

Задание: Спроектируйте высокоуровневую архитектуру промышленной ИИ-системы (MLOps). Опишите, как будет организован поток данных (использование брокеров сообщений, например, Kafka), как будет развернут сам ИИ-сервис (микросервисы, Docker, Kubernetes). Разработайте схему CI/CD пайплайна для машинного обучения, включающую автоматическое тестирование модели, версионирование (MLflow/DVC) и стратегию безопасного развертывания (например, Canary release или A/B тестирование) с возможностью быстрого отката.

Результат выполнения: Схема архитектуры системы (C4 model или UML) и описание регламента CI/CD пайплайна для ИИ-моделей.

МДК 01.02. Разработка мобильных приложений с поддержкой искусственного интеллекта

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

Кейс-задания 1.

Кейс-сценарий: Стартап планирует разработать MVP (минимально жизнеспособный продукт) мобильного приложения для трекинга физической активности с базовыми ИИ-функциями (распознавание упражнений по видео). У компании ограниченный бюджет, жесткий дедлайн (3 месяца) и небольшая

команда: один Android-разработчик (Kotlin), один iOS-разработчик (Swift) и один бэкенд-разработчик.

Задание: Проанализируйте варианты выбора технологического стека: нативная разработка (Kotlin/Swift) против кроссплатформенных решений (Flutter, React Native, Kotlin Multiplatform). Оцените риски и преимущества каждого подхода для данной команды и задачи. Предложите оптимальную архитектуру приложения (MVVM, MVI или Clean Architecture) и настройку системы контроля версий (GitFlow) для параллельной работы команды.

Кейс-задание 2.

Кейс-сценарий: Агротехническая компания разрабатывает приложение для агрономов, которое должно распознавать болезни сельскохозяйственных культур по фотографии листа. Главная проблема: поля часто находятся в зонах с плохим или полностью отсутствующим интернет-покрытием. Кроме того, приложение будет использоваться в полевых условиях по 6-8 часов, поэтому критически важно минимизировать расход батареи и нагрев устройства.

Задание: Спроектируйте стратегию интеграции ИИ-модели. Обоснуйте выбор между облачным API, локальным инференсом (TFLite/CoreML) или гибридной архитектурой. Предложите конкретные методы оптимизации модели (квантизация, прунинг) и настройки аппаратного ускорения (NNAPI, GPU Delegate) для достижения баланса между точностью распознавания, скоростью отклика и энергопотреблением.

Результат выполнения: Схема интеграции ИИ-модели с описанием методов оптимизации и оценкой компромиссов (trade-offs) по производительности и батарее.

Кейс-задание 3.

Кейс-сценарий: Музей заказывает разработку интерактивного мобильного гида. Приложение должно в реальном времени распознавать экспонаты через камеру (с наложением AR-меток) и генерировать адаптивный аудиорассказ с помощью NLP. Сложность заключается в том, что ИИ-модуль должен учитывать контекст: скорость ходьбы пользователя и время его задержки у экспоната, чтобы подстраивать длину и сложность рассказа.

Задание: Спроектируйте UX-поток (User Flow) и техническое взаимодействие между камерой, AR-движком, сенсорами устройства (акселерометр/гироскоп) и ИИ-модулем. Опишите, как организовать обработку непрерывного потока данных с сенсоров и камеры в реальном времени, чтобы интерфейс оставался отзывчивым (без фриз и лагов) и предоставлял пользователю мгновенную обратную связь.

Результат выполнения: Диаграмма потоков данных (Data Flow Diagram) для работы сенсоров и ИИ в реальном времени, а также описание UX-решений

для интерактивного взаимодействия.

Кейс-задание 4.

Кейс-сценарий: Команда готовит к релизу тяжелое мобильное приложение, которое включает локальную NLP-модель размером 250 МБ. Магазины приложений (Google Play и App Store) имеют строгие ограничения на размер первоначальной загрузки через мобильные сети (например, 150 МБ). Кроме того, отдел Data Science планирует обновлять языковую модель каждую неделю, и вынуждать пользователей скачивать полное обновление приложения (200+ МБ) каждую неделю недопустимо.

Задание: Разработайте стратегию развертывания и дистрибуции приложения. Опишите, как использовать механизмы динамической доставки (Play Feature Delivery / On-Demand Resources) для уменьшения размера APK/AAB. Предложите архитектуру системы динамического обновления ИИ-моделей без перевыпуска самого приложения (например, через Firebase Remote Config или собственный менеджер моделей). Определите ключевые метрики для мониторинга приложения после релиза.

Результат выполнения: План релиза и дистрибуции, схема механизма динамического обновления моделей и список метрик для дашборда мониторинга (Crashlytics, производительность).

Кейс-задание 5.

Кейс-сценарий: Финтех-компания внедряет в мобильное банковское приложение функцию подтверждения крупных переводов с помощью распознавания лица (Liveness detection + Face ID). Приложение локально генерирует и хранит биометрический эмбединг (векторное представление лица). Служба безопасности банка опасается атак на извлечение модели (Model Extraction), реверс-инжиниринга APK-файла и кражи локальных данных на рутированных (rooted/jailbroken) устройствах.

Задание: Спроектируйте архитектуру безопасности ИИ-модуля. Опишите механизмы безопасного хранения биометрических данных (Android Keystore / iOS Keychain). Предложите комплекс мер по защите самой ИИ-модели от кражи и несанкционированного доступа (обфускация кода, RASP, шифрование весов модели). Как приложение должно реагировать на обнаружение root-прав или эмулятора?

Результат выполнения: Модель угроз (Threat Model) для ИИ-модуля и подробный план мероприятий по обеспечению безопасности и защите интеллектуальной собственности.

Кейс-задание 6.

Кейс-сценарий: Мобильное приложение с ИИ-переводчиком находится на рынке уже год. Аналитика показывает падение удержания пользователей

(Retention Rate). Пользователи хвалят качество перевода, но жалуются на частые вылеты приложения на старых устройствах и непонятный интерфейс, когда ИИ «думает». Продакт-менеджер хочет внедрить новую платную ИИ-функцию (синхронный перевод голоса), но сначала нужно обосновать ее необходимость и исправить текущие проблемы.

Задание: Составьте план развития продукта (Product Roadmap) на следующий квартал. Опишите, как выстроить Agile-процессы в команде, чтобы одновременно закрывать технический долг (исправление крашей) и разрабатывать новую фичу. Спроектируйте А/В-тест для валидации спроса на новую платную функцию. Предложите UX-решения для улучшения взаимодействия пользователя с ИИ в моменты задержки обработки.

Результат выполнения: Дорожная карта продукта (Roadmap), дизайн А/В-теста с целевыми метриками и предложения по улучшению UX при работе с ИИ.

МДК 01.03. Тестирование программных модулей (ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.6, ПК 1.7)

Кейс-задание 1.

Кейс-сценарий: Крупная розничная сеть обратилась к вашей компании с запросом на внедрение технологий ИИ. У компании есть три проблемы: 1) падение продаж из-за нерелевантных предложений клиентам; 2) высокая нагрузка на колл-центр из-за типовых вопросов; 3) очереди на кассах в часы пик, которые приводят к оттоку покупателей. В распоряжении компании есть база данных покупок, логи чатов и звонков, а также видеокамеры в торговых залах.

Задание: Проанализируйте бизнес-задачи клиента. Предложите по одному решению на базе ИИ для каждой из трех проблем. Для каждого решения обоснуйте выбор направления ИИ (машинное обучение, обработка естественного языка, компьютерное зрение), опишите, какие именно данные потребуются для обучения моделей, и выявите потенциальные этические риски и риски приватности при внедрении данных систем.

Результат выполнения: Аналитическая записка с матрицей «Проблема — Решение — Направление ИИ — Требуемые данные — Риски».

Кейс-задание 2.

Кейс-сценарий: Медицинский стартап разрабатывает систему прогнозирования рисков сердечно-сосудистых заболеваний. Данные поступают из трех разных клиник в разных форматах: CSV-таблицы с показателями анализов (где 30% значений пропущено), неструктурированные текстовые заметки врачей и изображения ЭКГ. Кроме того, заболевание

встречается лишь у 5% пациентов в выборке. Юридический отдел требует соблюдения строгих норм обезличивания данных.

Задание: Вы выступаете в роли Lead Data Engineer. Спроектируйте конвейер (pipeline) подготовки данных. Опишите стратегию сбора и безопасного хранения данных. Предложите конкретные методы обработки пропусков для числовых и текстовых данных. Разработайте план действий по работе с сильным дисбалансом классов (5% против 95%) и предложите 3 идеи для Feature Engineering на основе медицинских показателей.

Результат выполнения: Схема конвейера данных (Data Pipeline) и техническое обоснование выбранных методов предобработки.

Кейс-задание 3.

Кейс-сценарий: Банк поручает вашей команде разработать модель скоринга для одобрения кредитов. Руководство банка требует, чтобы модель была максимально точной. Однако юридический департамент выдвигает жесткое условие: модель должна быть интерпретируемой. При отказе в кредите менеджер должен иметь возможность объяснить клиенту конкретные причины (например, «отказ из-за высокой долговой нагрузки и отсутствия кредитной истории»), а не просто сослаться на «решение черного ящика».

Задание: Проведите сравнительный анализ трех алгоритмов: Логистическая регрессия, Случайный лес (Random Forest) и Градиентный бустинг (XGBoost/LightGBM) в контексте данного ТЗ. Выберите оптимальный алгоритм, обоснуйте свой выбор с точки зрения баланса между точностью и интерпретируемостью. Опишите, как вы будете кодировать категориальные признаки (тип занятости, семейное положение) для выбранной модели.

Результат выполнения: Сравнительная таблица алгоритмов и обоснованный выбор финального решения для защиты перед заказчиком.

Кейс-задание 4.

Кейс-сценарий: Телеком-компания разработала модель для прогнозирования оттока клиентов (Churn Rate). На тестовой выборке модель показала Accuracy (точность) 94%. Руководство было радо, но после внедрения маркетинговый отдел сообщил, что модель бесполезна: она почти не находит тех, кто реально уходит, и тратит бюджет на удержание лояльных клиентов. В выборке 94% клиентов остаются и 6% уходят.

Задание: Объясните руководству, почему метрика Accuracy ввела их в заблуждение (парадокс точности). Определите, какие метрики (Precision, Recall, F1-score, ROC-AUC) необходимо использовать в данном случае, и обоснуйте свой выбор с точки зрения бизнес-логики (что для компании дороже: ложноположительное или ложноотрицательное срабатывание?). Предложите 3 конкретных метода (на уровне данных или алгоритма) для

улучшения способности модели находить «уходящих» клиентов.

Результат выполнения: Отчет об анализе ошибок модели с расчетом бизнес-стоимости ошибок и планом по улучшению метрики Recall.

Кейс-задание 5.

Кейс-сценарий: Агротехническая компания внедряет дрон для мониторинга состояния посевов. Дрон должен в реальном времени распознавать болезни растений по фотографиям листьев. Проблема: текущая сверточная нейронная сеть (CNN) показывает высокую точность на сервере, но она слишком «тяжелая» — время инференса составляет 2 секунды, а батарея дрона садится за 10 минут. При этом размеченных фотографий болезней растений у компании всего 5 000 штук.

Задание: Вам необходимо оптимизировать решение для работы на Edge-устройстве (бортовом компьютере дрона). Предложите комплексную стратегию решения проблемы. Во-первых, как повысить качество модели при малом объеме данных (аугментация, трансферное обучение)? Во-вторых, какие методы оптимизации и сжатия модели (квантизация, прунинг, выбор легкой архитектуры) вы примените для ускорения инференса и снижения потребления ресурсов без критической потери точности?

Результат выполнения: Техническое задание на оптимизацию нейросети с описанием выбранных архитектурных решений и методов сжатия.

Кейс-задание 6.

Кейс-сценарий: Финтех-компания запускает систему обнаружения мошеннических транзакций (Fraud Detection) в реальном времени. Система должна обрабатывать до 10 000 транзакций в секунду с задержкой (latency) не более 50 мс. Кроме того, команда Data Science хочет иметь возможность обновлять модели машинного обучения каждую неделю без остановки сервиса (Zero Downtime), а в случае падения качества новой модели — мгновенно откатываться на предыдущую версию.

Задание: Спроектируйте высокоуровневую архитектуру промышленной ИИ-системы (MLOps). Опишите, как будет организован поток данных (использование брокеров сообщений, например, Kafka), как будет развернут сам ИИ-сервис (микросервисы, Docker, Kubernetes). Разработайте схему CI/CD пайплайна для машинного обучения, включающую автоматическое тестирование модели, версионирование (MLflow/DVC) и стратегию безопасного развертывания (например, Canary release или A/B тестирование) с возможностью быстрого отката.

Результат выполнения: Схема архитектуры системы (C4 model или UML) и описание регламента CI/CD пайплайна для ИИ-моделей.

Критерии оценки выполнения кейс-заданий

Оценка «отлично» – кейс–задание выполнено полностью, в рамках регламента, установленного на публичную презентацию, студент(ы) приводит (подготовили) полную четкую аргументацию выбранного решения на основе качественно сделанного анализа. Демонстрируются хорошие теоретические знания, имеется собственная обоснованная точка зрения на проблему(ы) и причины ее (их) возникновения. В случае ряда выявленных проблем четко определяет их иерархию. При устной презентации уверенно и быстро отвечает на заданные вопросы, выступление сопровождается приемами визуализации. В случае письменного отчета-презентации по выполнению кейс-задания сделан структурированный и детализированный анализ кейса, представлены возможные варианты решения (3-5), четко и аргументировано обоснован окончательный выбор одного из альтернативных решений.

Оценка «хорошо» – кейс–задание выполнено полностью, но в рамках установленного на выступление регламента, студент(ы) не приводит (не подготовили) полную четкую аргументацию выбранного решения. Имеет место излишнее теоретизирование, или наоборот, теоретическое обоснование ограничено, имеется собственная точка зрения на проблемы, но не все причины ее возникновения установлены. При устной презентации на дополнительные вопросы выступающий отвечает с некоторым затруднением, подготовленная устная презентация выполненного кейс-задания не очень структурирована. При письменном отчете-презентации по выполнению кейс задания сделан не полный анализ кейса, без учета ряда фактов, выявлены не все возможные проблемы, для решения могла быть выбрана второстепенная, а не главная проблема, количество представленных возможных вариантов решения – 2-3, затруднена четкая аргументация окончательного выбора одного из альтернативных решений.

Оценка «удовлетворительно» – кейс–задание выполнено более чем на 2/3, но в рамках установленного на выступление регламента, студент(ы) расплывчато раскрывает решение, не может четко аргументировать сделанный выбор, показывает явный недостаток теоретических знаний. Выводы слабые, свидетельствуют о недостаточном анализе фактов, в основе решения может иметь место интерпретация фактов или предположения, Собственная точка зрения на причины возникновения проблемы не обоснована или отсутствует. При устной презентации на вопросы отвечает с трудом или не отвечает совсем. Подготовленная презентация выполненного кейс-задания не структурирована. В случае письменной презентации по выполнению кейс задания не сделан детальный анализ кейса, далеко не все факты учтены, для решения выбрана второстепенная, а не главная проблема,

количество представленных возможных вариантов решения – 1-2, отсутствует четкая аргументация окончательного выбора решения.

Оценка «неудовлетворительно» – кейс-задание не выполнено, или выполнено менее чем на треть. Отсутствует детализация при анализе кейса, изложение устное или письменное не структурировано. Если решение и обозначено в выступлении или отчете-презентации, то оно не является решением проблемы, которая заложена в кейсе.

Виды работ и задания по учебной практике

Учебная практика раздела 1

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

Виды работ по учебной практике раздела 1

1. Сбор и предобработка данных из открытых источников.
2. Разработка программных модулей для анализа данных.
3. Разработка базовых моделей машинного обучения.
4. Визуализация результатов работы моделей ИИ.
5. Написание юнит-тестов для ИИ-модулей

Задания по учебной практике раздела 1

Задание 1. Осуществить сбор данных из открытого источника (с использованием веб-скрапинга или API). Выполнить первичный анализ и визуализацию датасета. Провести очистку данных от пропусков и выбросов, нормализацию и стандартизацию. Применить методы Feature Engineering для создания новых признаков и подготовить несбалансированные данные для дальнейшего обучения модели.

Задание 2. Реализовать прототип базовой модели ИИ для классификации данных (или базовой системы рекомендаций) с использованием стандартных библиотек, проанализировать результаты и оформить отчет о проделанной работе.

Задание 3. Разделить исходный датасет на обучающую и тестовую выборки. Провести кросс-валидацию (включая стратифицированную и валидацию на временных рядах). Оценить качество обученной модели путем построения матрицы ошибок (Confusion Matrix), ROC-кривой и кривых Precision-Recall. Выполнить настройку гиперпараметров с использованием GridSearchCV и применить L1/L2-регуляризацию для предотвращения переобучения и улучшения алгоритма.

Задание 4. Спроектировать и визуализировать архитектуру нейронной

сети (сверточной CNN для задачи распознавания изображений или рекуррентной LSTM для анализа временных рядов). Исследовать влияние различных функций активации (ReLU, сигмоида) и оптимизаторов на процесс обучения. Применить методы аугментации данных, пакетную нормализацию и трансферное обучение (Transfer learning) с использованием предобученных сетей для повышения точности модели. Сохранить обученную модель для последующего развертывания.

Задание 5. Спроектировать микросервисную архитектуру для ИИ-системы. Выполнить контейнеризацию разработанной ИИ-модели с использованием Docker и настроить ее развертывание. Создать REST API для интеграции ИИ-сервиса с внешними системами. Настроить процесс версионирования моделей и данных (с использованием MLflow или DVC), а также реализовать базовый CI/CD пайплайн для автоматического тестирования и обновления модели. Провести нагрузочное тестирование системы и настроить мониторинг ее надежности.

Учебная практика раздела 2

(ОК 01; ОК 02, ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

Виды работ по учебной практике раздела 2

1. Установка и настройка среды разработки.
2. Разработка пользовательского интерфейса.
3. Интеграция предобученной модели в мобильное приложение.
4. Разработка прототипа мобильного приложения с ИИ.
5. Работа с системами контроля версий.

Задания по учебной практике раздела 2

Задание 1: Создание базового каркаса мобильного приложения. Настройка среды Android Studio и конфигурации Gradle. Разработка многоэкранного пользовательского интерфейса с использованием Navigation Component, RecyclerView и ConstraintLayout. Реализация обработки событий взаимодействия с пользователем, внедрение темной темы и поддержка базовых требований доступности (Accessibility).

Результат выполнения: Работающий прототип Android-приложения с настроенной навигацией между экранами, адаптивной версткой и корректной обработкой UI-событий.

Задание 2. Разработка интерактивного модуля приложения с использованием встроенных сенсоров и ИИ. Создание голосового помощника

или модуля распознавания жестов с реализацией мгновенной визуальной и звуковой обратной связи для пользователя. Настройка обработки естественного языка (NLP) или интеграция AR-компонентов для наложения виртуальных объектов на реальный мир.

Результат выполнения: Интерактивный модуль приложения, использующий микрофон, камеру или акселерометр в связке с ИИ-алгоритмами для обеспечения бесшовного взаимодействия с пользователем.

Задание 3. Интеграция предобученной TensorFlow Lite модели в Android-приложение. Подключение камеры через библиотеку CameraX для захвата видеопотока и передачи кадров в ИИ-модель в реальном времени (например, для классификации изображений или обнаружения объектов). Применение квантизации модели для уменьшения ее размера и настройка аппаратного ускорения инференса (GPU Delegate).

Результат выполнения: Мобильное приложение, способное в реальном времени анализировать изображение с камеры устройства с использованием оптимизированной локальной ИИ-модели.

Задание 4: Подготовка мобильного ИИ-приложения к релизу. Настройка CI/CD пайплайна для автоматического запуска тестов. Оптимизация размера итоговой сборки (формат AAB), выполнение криптографической подписи приложения и настройка мониторинга сбоев (Crashlytics) и ошибок ANR после выпуска версии в Google Play Console.

Результат выполнения: Подписанная и оптимизированная сборка приложения, готовая к публикации, с настроенным пайплайном непрерывной интеграции и подключенными системами пост-релизной аналитики.

Задание 5: Реализация подсистемы безопасности мобильного приложения. Настройка биометрической аутентификации и шифрование локальных данных. Обеспечение безопасного хранения API-ключей и защита сетевого трафика. Внедрение методов обфускации кода и защиты файла ИИ-модели от несанкционированного извлечения и реверс-инжиниринга.

Результат выполнения: Защищенный модуль приложения с реализованной аутентификацией, шифрованием данных и мерами по предотвращению кражи интеллектуальной собственности (ИИ-модели).

Учебная практика раздела 3

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.6, ПК 1.7)

Виды работ по учебной практике раздела 3

1. Разработка тестовых сценариев.
2. Ручное тестирование ИИ-модулей.
3. Автоматизация тестирования.

4. Тестирование производительности.
5. Документирование результатов тестирования.

Задания по учебной практике раздела 3

Задание 1: Разработка тестовой стратегии и пакета тестовых артефактов для ИИ-модуля. Студент должен создать тестовый план, разработать чек-листы для смоук-тестирования ИИ-сервиса и подготовить детальные тестовые сценарии (тест-кейсы) с использованием стандартных шаблонов. В конце необходимо составить матрицу трассировки для оценки покрытия тестами бизнес-требований.

Результат выполнения: Комплект документации: Тестовый план, чек-листы, пакет тестовых сценариев и матрица трассировки требований.

Задание 2: Разработка автоматизированных тестов и настройка CI/CD пайплайна для ИИ-модуля. Студент должен написать юнит-тесты для функций предобработки данных, создать моки (mocks) и стабы (stubs) для изоляции внешних API-зависимостей ИИ-сервиса, а также настроить скрипт автоматизации регрессионного тестирования с генерацией отчета в рамках CI/CD пайплайна.

Результат выполнения: Исходный код автоматизированных тестов (с использованием мок-объектов) и конфигурационный файл CI/CD пайплайна с шагом запуска тестов и генерации отчета.

Задание 3: Настройка тестового окружения и тестирование API ИИ-сервиса. Студент должен подготовить тестовый стенд с использованием заглушек для имитации базы данных и внешних сервисов, разработать коллекцию запросов в Postman/SoapUI для тестирования REST API ИИ-сервиса (проверка валидации входных данных, обработки исключений, проверки формата JSON и времени отклика) и провести комплексное приемочное тестирование перед релизом.

Результат выполнения: Коллекция API-тестов (Postman/SoapUI), описание настроенного тестового окружения со заглушками и отчет о результатах приемочного тестирования API.

Задание 4: Проектирование тестов с использованием спецификационных методов «черного ящика». Студенту необходимо применить метод эквивалентного разбиения и анализа граничных значений для входных параметров ИИ-модели, составить таблицу решений для сложной логики классификатора (с множеством условий) и разработать сценарии тестирования на основе диаграмм Use Case.

Результат выполнения: Набор тестовых сценариев, разработанных с применением техник эквивалентного разбиения, таблиц решений и Use Case,

с расчетом процента покрытия бизнес-логики.

Задание 5: Разработка модели управления дефектами и построение диаграммы причинно-следственных связей. Студенту необходимо классифицировать виды тестирования для заданного ИИ-проекта, описать жизненный цикл дефекта с учетом специфики ИИ (например, учет дрейфа данных), а также построить диаграмму причинно-следственных связей для сложного бизнес-правила системы и вывести на ее основе первичные тестовые условия.

Результат выполнения: Документ с классификацией тестов, регламент управления дефектами (с матрицей серьезности и приоритета) и графическая диаграмма причинно-следственных связей с вытекающими тестовыми условиями.

Производственная практика (по профилю специальности) раздела 1 (ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

Виды работ и задания по производственной практике раздела 1

1. Сбор и обработка больших объемов данных для обучения моделей ИИ.
2. Feature engineering и создание признаков.
3. Проектирование и реализация моделей машинного обучения.
4. Разработка моделей глубокого обучения.
5. Оптимизация моделей ИИ для повышения производительности
6. Валидация и тестирование моделей.
7. Разработка пайплайна машинного обучения.

Производственная практика (по профилю специальности) раздела 2 (ОК 01; ОК 2; ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

Виды работ и задания по производственной практике раздела 2

1. Проектирование архитектуры мобильного ИИ-приложения.
2. Разработка сложных ИИ-приложений для мобильных платформ.
3. Интеграция ИИ-моделей в существующие информационные системы.
4. Оптимизация ИИ-моделей для мобильных устройств.
5. Разработка и публикация мобильных приложений.
6. Внедрение систем аналитики и мониторинга.
7. Обеспечение безопасности мобильного приложения.
8. Отладка мобильного ИИ-приложения.

Производственная практика (по профилю специальности) раздела 3 (ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.6, ПК 1.7)

Виды работ и задания по производственной практике раздела 3

1. Разработка стратегии тестирования ИИ-систем.
2. Автоматизация тестирования программных продуктов.
3. Проведение интеграционного тестирования сложных систем ИИ.
4. Тестирование качества ИИ-моделей.
5. Нагрузочное и стресс-тестирование.
6. Тестирование безопасности ИИ-приложений.
7. Мониторинг производительности в реальных условиях.
8. Разработка и внедрение систем автоматизированного развертывания.
9. Документирование процесса тестирования

Критерии оценки выполнения обучающимся программы практики

Оценка «отлично» выставляется обучающемуся если выполнено 86-100 % заданий, предусмотренных в индивидуальном задании на практику, в период прохождения практики отмечена сформированность всех знаний и умений, предусмотренных программой практики.

Оценка «хорошо» выставляется обучающемуся если выполнено 70-85 % заданий, предусмотренных в индивидуальном задании на практику; задания выполнены с отдельными погрешностями, что повлияло на качество полученных результатов; в период прохождения практики отмечена сформированность основных знаний, и умений, предусмотренных программой практики.

Оценка «удовлетворительно» выставляется обучающемуся если выполнено 50-69% заданий, предусмотренных в индивидуальном задании на практику; структура отчета не в полной мере соответствует рекомендуемой; в период прохождения практики отмечена сформированность не менее 50% знаний, умений и навыков, предусмотренных программой практики.

Оценка «неудовлетворительно» выставляется обучающемуся если выполнено менее 50% заданий, предусмотренных в индивидуальном задании на практику; в период прохождения практики отмечена несформированность знаний, умений и навыков, предусмотренных программой практики.

2. Вопросы и задания для промежуточной аттестации

Экзаменационные вопросы и задания для устного (письменного) экзамена по междисциплинарному курсу

МДК 01.01. Разработка программных модулей в системах искусственного интеллекта
(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

Теоретические вопросы для устного (письменного) экзамена по МДК 01.01.

1. История и эволюция искусственного интеллекта. Основные направления ИИ: машинное обучение, глубокое обучение, нейронные сети. Этические вызовы ИИ.
2. Методы сбора данных для обучения ИИ-моделей: веб-скрапинг, API, базы данных. Правовые и технические аспекты.
3. Методы предобработки данных: очистка от пропусков и выбросов, нормализация, стандартизация и кодирование категориальных данных.
4. Feature engineering: принципы создания новых признаков. Методы работы с несбалансированными данными. Валидация и контроль качества данных.
5. Виды обучения: с учителем, без учителя, с подкреплением. Сравнительная характеристика и области применения.
6. Основные алгоритмы классификации и регрессии: линейная и логистическая регрессия, метод k-ближайших соседей (kNN), метод опорных векторов (SVM), наивный байесовский классификатор.
7. Деревья решений и ансамблевые методы (случайный лес, градиентный бустинг). Принципы работы и анализ.
8. Кластеризация (k-means, агломеративная кластеризация) и уменьшение размерности (метод главных компонент - PCA). Поиск ассоциативных правил (Apriori).
9. Методы оценки качества моделей классификации: матрица ошибок (Confusion Matrix), Accuracy, Precision, Recall, F1-score. ROC-кривая и AUC.
10. Валидация моделей: разделение данных на обучающую и тестовую выборки, кросс-валидация (стандартная, стратифицированная, вложенная, на временных рядах).
11. Регуляризация моделей (L1 и L2). Оптимизация гиперпараметров и калибровка вероятностей модели.

12. Архитектуры нейронных сетей: многослойные перцептроны (MLP), сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN).
13. Процессы обучения нейронных сетей: обратное распространение ошибки, стохастический градиентный спуск (SGD), функции активации (ReLU, сигмоидальная).
14. Методы улучшения обучения нейросетей: пакетная нормализация (Batch Normalization), регуляризация (Dropout), аугментация данных.
15. Применение нейросетей: LSTM для последовательностей, автоэнкодеры для сжатия данных, трансферное обучение (Transfer learning). Сохранение и развертывание нейросети.
16. Принципы проектирования архитектуры ИИ-систем: модульность, масштабируемость, микросервисная архитектура, проектирование REST API.
17. Контейнеризация ИИ-модели с использованием Docker и развертывание в Kubernetes.
18. Настройка CI/CD пайплайна для машинного обучения. Версионирование моделей и данных (MLflow, DVC).
18. Обеспечение безопасности, мониторинг и надежность ИИ-систем. Интеграция очередей сообщений (RabbitMQ/Kafka). Стратегии резервного копирования и A/B тестирование моделей.

Практические ситуационные задания для устного (письменного) экзамена по МДК 01.01.

Задание 1. Проектирование конвейера данных

Ситуация: Вам предоставлен «сырой» дата сет для задачи прогнозирования оттока клиентов телеком-оператора. В данных присутствуют пропуски в числовых и категориальных признаках, выбросы в столбце «доход», а целевой класс «ушел» составляет лишь 5% выборки.

Задание: Разработайте алгоритм и опишите пошаговый конвейер (pipeline) предобработки данных. Укажите, какие методы вы используете для заполнения пропусков, обработки выбросов и кодирования категориальных признаков. Предложите два метода решения проблемы несбалансированности классов.

Задание 2. Выбор алгоритма и метрик оценки (ПК 1.1, ПК 1.2, ПК 1.6)

Ситуация: Банку необходима модель для выявления мошеннических транзакций (Fraud Detection). Стоимость пропуска мошеннической операции (ложноотрицательное срабатывание) для банка в 100 раз выше, чем стоимость ложной блокировки карты честного клиента (ложноположительное срабатывание).

Задание: Обоснуйте выбор алгоритма машинного обучения для данной

задачи. Определите, какие метрики качества (Accuracy, Precision, Recall, F1-score, ROC-AUC) будут являться целевыми для оптимизации модели. Нарисуйте схему матрицы ошибок и объясните, как вы будете настраивать порог отсечения (threshold) модели.

Задание 3. Архитектура нейронной сети (ПК 1.2, ПК 1.3)

Ситуация: Агротехнической компании требуется мобильное приложение, которое по фотографии листа растения определяет наличие болезни. Датасет содержит 10 000 размеченных изображений.

Задание: Спроектируйте архитектуру сверточной нейронной сети (CNN) для решения этой задачи. Опишите назначение сверточных и пулинговых слоев. Выберите подходящие функции активации и оптимизатор. Объясните, как вы будете применять трансферное обучение и аугментацию данных для повышения качества модели на относительно небольшой выборке.

Задание 4. Развертывание и CI/CD (MLOps) (ПК 1.4, ПК 1.5, ПК 1.7)

Ситуация: Команда Data Science обучила новую версию модели рекомендаций, которая показывает прирост метрики F1-score на 4% по сравнению с текущей продакшн-моделью. Модель необходимо внедрить в высоконагруженный микросервис без остановки сервиса (Zero Downtime).

Задание: Опишите процесс контейнеризации модели с помощью Docker. Спроектируйте схему CI/CD пайплайна для автоматического тестирования и развертывания модели. Опишите, как вы организуете версионирование модели и данных (с использованием MLflow/DVC) и какую стратегию развертывания (Canary release или A/B тестирование) выберете для безопасного внедрения.

Задание 5. Командная разработка и отладка (ПК 1.4, ПК 1.5)

Ситуация: В процессе групповой разработки ИИ-модуля два разработчика одновременно изменили один и тот же файл конфигурации и файл с функциями предобработки данных. При попытке слияния веток (merge) возник конфликт. Кроме того, в продакшн-среде модуль начал выдавать ошибку OutOfMemoryError при обработке больших батчей данных.

Задание: Опишите алгоритм разрешения конфликта слияния в системе контроля версий (Git) с использованием pull request и code review. Предложите методы логирования и профилирования производительности (с использованием специализированных средств) для выявления и устранения причины утечки памяти.

Критерии оценки ответов на экзаменационные вопросы, выполнения экзаменационных заданий по МДК 01.01

Оценка «отлично»: теоретическое содержание модуля освоено полностью.

Практические навыки сформированы. Задания выполнены полностью, соответствуют поставленной задаче (алгоритмы и модели разработаны в полном соответствии с ТЗ, оптимизированы, код оформлен по стандартам, тесты успешны). Ошибки и недочеты отсутствуют.

Оценка «хорошо»: теоретическое содержание освоено полностью. Практические навыки сформированы не в полном объеме. Задания выполнены, но допущены незначительные ошибки и недочеты (например, алгоритмы разработаны в соответствии с ТЗ, но допускают незначительные отклонения в оптимизации; код оформлен с мелкими недочетами; тесты в целом успешны, но не покрывают все edge-cases).

Оценка «удовлетворительно»: теоретическое содержание освоено частично (пробелы не носят существенного характера). Практические навыки сформированы частично. Задания выполнены с существенными доработками (алгоритмы и модули разработаны с частичным соответствием ТЗ, код оформлен не в полном соответствии со стандартами, тестирование выполнено частично, ошибки выявлены, но не все исправлены).

Оценка «неудовлетворительно»: теоретическое содержание не освоено. Практические навыки не сформированы. Выполненные задания содержат грубые ошибки и недочеты, алгоритмы не соответствуют ТЗ, код не работает или не оформлен, тестовые сценарии не составлены.

Вопросы для устного (письменного) зачета по междисциплинарному курсу

МДК 01.02. Разработка мобильных приложений с поддержкой
искусственного интеллекта

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5)

Теоретические вопросы (для устного/письменного зачета)

1. Опишите архитектуру Android-приложения и роль файла конфигурации build.gradle. В чем разница между форматами APK и AAB?
2. Сравните языки Kotlin и Java для мобильной разработки. Опишите преимущества Kotlin и его особенности (coroutines, null safety).
3. Объясните принципы адаптивной верстки в Android. Назначение ConstraintLayout, RecyclerView и Navigation Component.
4. Опишите пошаговый процесс внедрения модели машинного обучения в мобильное приложение с использованием TensorFlow Lite.
5. Сравните локальный инференс (Edge AI) и облачные AI API. В каких случаях целесообразно использовать гибридную архитектуру (онлайн/офлайн)?

6. Объясните физическую и математическую суть методов оптимизации ИИ-моделей для мобильных устройств: квантизации (Quantization) и прунинга (Pruning).
7. Какую роль играет Neural Networks API (NNAPI) и GPU Delegate в Android? Как они влияют на латентность и потребление батареи?
8. Опишите процесс интеграции камеры (CameraX) для непрерывной передачи видеопотока в ИИ-модель в реальном времени.
9. Какие сенсоры мобильного устройства можно использовать для сбора контекстных данных и взаимодействия с ИИ-модулями?
10. Опишите архитектурные принципы создания голосового помощника или модуля распознавания жестов с мгновенной обратной связью.
11. Опишите процесс подготовки мобильного приложения к релизу: оптимизация размера, криптографическая подпись, публикация в Google Play Console.
12. Как настроить CI/CD пайплайн для мобильного приложения? Какую роль играет GitFlow и Code Review в командной разработке?
13. Где и как в Android-приложении необходимо безопасно хранить конфиденциальные данные (API-ключи, токены, биометрические эмбединги)?
14. Какие существуют специфические угрозы безопасности для ИИ-моделей на мобильных устройствах? Опишите методы защиты модели от кражи и реверс-инжиниринга.
15. Объясните методологию проведения A/B тестирования новых ИИ-функций в приложении. Какие метрики необходимо собирать для принятия решения о раскатке?
16. Опишите подходы к пост-релизному мониторингу приложения с использованием Firebase Crashlytics и аналитики ANR (Application Not Responding).

Критерии оценки ответов на вопросы для устного (письменного) зачета по МДК 01.02

Оценка «отлично»: студент демонстрирует глубокое понимание жизненного цикла мобильного ИИ-приложения, платформенных особенностей (Android/iOS) и механизмов интеграции нейросетей. Ответы исчерпывающие, логичные, структурированные. Студент свободно оперирует профессиональной терминологией (TensorFlow Lite, CameraX, NNAPI, GPU Delegate, квантизация, прунинг, AAB, GitFlow, MVVM, Crashlytics и др.). Способен проводить самостоятельный сравнительный анализ (например, локальный Edge AI vs облачные API, нативная vs кроссплатформенная

разработка), четко обосновывать выбор архитектурных решений и методов оптимизации моделей. Глубокое понимание вопросов безопасности мобильных ИИ-приложений (защита весов модели, биометрия, Keystore) и пост-релизной аналитики.

Оценка «хорошо»: студент уверенно отвечает на основные вопросы билета или списка. Допускаются незначительные неточности, шероховатости в формулировках или мелкие фактические ошибки, которые студент легко исправляет после замечания преподавателя (например, неточности в математическом обосновании квантизации или мелкие детали настройки CI/CD пайплайна). Терминология используется в целом верно, но возможны небольшие затруднения при объяснении специфических низкоуровневых механизмов (например, точные принципы работы делегатов NNAPI или детали статистической значимости A/B тестов). В целом понимает разницу между форматами сборок (APK/AAB), методы оптимизации и требования безопасности, но не всегда может аргументированно выбрать оптимальное решение для нестандартной бизнес-задачи без наводящих вопросов.

Оценка «удовлетворительно»: студент знает базовые понятия мобильной разработки и основы интеграции ИИ. Ответы фрагментарны. Студент испытывает трудности при ответе на вопросы, требующие глубокого понимания архитектуры и оптимизации (например, путается в паттернах проектирования, не может четко описать процесс конвертации модели в TFLite). Допускаются ошибки в терминологии, путаница между смежными понятиями (например, подмена понятий локального инференса и облачного API, непонимание разницы между прунингом и квантизацией). Для раскрытия темы и формирования полного ответа требуются существенные наводящие вопросы преподавателя. Понимание специфики безопасности и CI/CD поверхностное.

Оценка «неудовлетворительно»: студент не может ответить на базовые вопросы по платформам мобильной разработки, инструментам (Android Studio, Gradle, Kotlin/Java) и основам интеграции ИИ-моделей. Отсутствует понимание ключевых концепций модуля (что такое TFLite, зачем нужна оптимизация моделей, как устроен жизненный цикл мобильного приложения). Допускаются грубые фактические ошибки, терминологическая путаница. Ответы не соответствуют профилю специальности, студент не понимает разницы между традиционной мобильной разработкой и разработкой приложений с поддержкой ИИ.

Вопросы для устного (письменного) зачета по междисциплинарному курсу

МДК 01.03. Тестирование программных модулей
(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.6, ПК 1.7)

Теоретические вопросы для устного (письменного) зачета

1. Дайте определение качества программного обеспечения в соответствии с ГОСТ Р ИСО/МЭК 25051. Перечислите основные метрики качества.
2. Опишите уровни тестирования (модульное, интеграционное, системное, приемочное) и их цели.
3. Что такое жизненный цикл дефекта? Как определяются серьезность (severity) и приоритет (priority) дефекта?
4. Чем функциональное тестирование отличается от нефункционального? Приведите примеры нагрузочного и стресс-тестирования.
5. Какова структура и назначение тестового плана и тестовой стратегии?
6. Опишите назначение чек-листов, тестовых пакетов и шаблонов тестовых сценариев.
7. Как проводится оценка покрытия тестами бизнес-требований и функциональных спецификаций?
8. Объясните суть техник «черного ящика»: метода эквивалентного разбиения и анализа граничных значений. Приведите примеры.
9. Для чего используются таблицы решений и как их применять для тестирования сложной бизнес-логики?
10. Опишите технику тестирования переходов состояний (State Transition Testing) и диаграммы причинно-следственных связей.
11. Что такое покрытие кода (code coverage)? В чем разница между покрытием строк и покрытием ветвей?
12. Опишите структуру матрицы ошибок (Confusion Matrix). Дайте определение метрикам Accuracy, Precision, Recall и F1-score.
13. Объясните физический смысл ROC-кривой и метрики AUC. Как анализировать пороговые значения классификатора?
14. Перечислите основные метрики оценки моделей регрессии (MAE, RMSE, MAPE) и кластеризации.
15. Что такое компромисс между смещением и дисперсией (Bias-Variance Tradeoff)? Как оценить оптимальную сложность модели ИИ?
16. Какова роль моков (mocks) и стабов (stubs) при тестировании ИИ-систем с внешними зависимостями?
17. Опишите процесс параметризации тестов и генерации синтетических тестовых данных для моделей машинного обучения.
18. Как интегрировать автоматизированное тестирование ИИ-моделей и пайплайнов данных в CI/CD пайплайн?

- 19.Что такое E2E (End-to-End) тестирование? Как настраиваются тестовые стенды и заглушки?
- 20.Опишите особенности кросс-платформенного тестирования ИИ-интерфейсов и оценки UX (включая интерпретируемость решений ХАИ).
- 21.Как проводится А/В тестирование ИИ-моделей в production-среде и анализ бизнес-метрик?
- 22.Опишите подходы к тестированию API ИИ-сервисов, а также к нагрузочному тестированию и тестированию безопасности.

Критерии оценки ответов на вопросы для устного (письменного) зачета по МДК 01.03

Оценка «отлично»: студент демонстрирует глубокое понимание жизненного цикла тестирования, методов тест-дизайна и метрик оценки ИИ-моделей. Ответы исчерпывающие, логичные, структурированные. Студент свободно оперирует профессиональной терминологией. Способен проводить самостоятельный сравнительный анализ, четко обосновывать выбор техник тест-дизайна для конкретных бизнес-задач. Глубокое понимание вопросов автоматизации тестирования, генерации синтетических данных, мониторинга дрейфа данных (Data Drift) и интерпретируемости решений ИИ (ХАИ).

Оценка «хорошо»: студент уверенно отвечает на основные вопросы билета или списка. Допускаются незначительные неточности, шероховатости в формулировках или мелкие фактические ошибки, которые студент легко исправляет после замечания преподавателя. Терминология используется в целом верно, но возможны небольшие затруднения при объяснении специфических низкоуровневых механизмов. В целом понимает разницу между уровнями тестирования, методами тест-дизайна и требованиями к тестированию ИИ, но не всегда может аргументированно выбрать оптимальную стратегию тестирования для нестандартной задачи без наводящих вопросов.

Оценка «удовлетворительно»: теоретическое содержание освоено частично, но пробелы не носят существенного (критического) характера. Студент знает базовые понятия тестирования и основы оценки ИИ-моделей. Ответы фрагментарны. Студент испытывает трудности при ответе на вопросы, требующие глубокого понимания архитектуры тестирования и специфики ИИ (например, путается в видах тест-дизайна, не может четко описать процесс расчета F1-score или настройки тестового окружения с заглушками). Допускаются ошибки в терминологии, путаница между смежными понятиями (например, подмена понятий «серьезность» (severity)

и «приоритет» (priority) дефекта, непонимание разницы между регрессионным и интеграционным тестированием).

Оценка «неудовлетворительно»: теоретическое содержание не освоено. Студент не может ответить на базовые вопросы по основам тестирования, инструментам, методам тест-дизайна и метрикам оценки ИИ-систем. Отсутствует понимание ключевых концепций модуля (что такое тестовый план, матрица ошибок, зачем нужна автоматизация). Допускаются грубые фактические ошибки, терминологическая путаница. Ответы не соответствуют профилю специальности, студент не понимает разницы между традиционным тестированием и тестированием ИИ-систем, не владеет основами проектирования тестовых сценариев.

Примерные вопросы для защиты отчета по учебной практике

ПМ.01 Разработка кода для обучения искусственного интеллекта
(ОК 01; ОК 2; ОК 04; ОК 05; ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5; ПК 1.6; ПК 1.7)

1. Как происходит сбор и предобработка данных из открытых источников.
2. Изложите порядок разработки программных модулей для анализа данных.
3. Как происходит разработка базовых моделей машинного обучения.
4. Как осуществляется визуализация результатов работы моделей ИИ.
5. Каков порядок написания юнит-тестов для ИИ-модулей
6. Как происходит установка и настройка среды разработки.
7. Опишите процесс разработки пользовательского интерфейса.
8. Как осуществляется интеграция предобученной модели в мобильное приложение.
9. Опишите порядок разработки прототипа мобильного приложения с ИИ.
10. Как осуществляется работа с системами контроля версий.
11. Опишите процесс разработки тестовых сценариев.
12. Каков порядок ручного тестирования ИИ-модулей.
13. Как осуществляется автоматизация тестирования.
14. Как происходит тестирование производительности.
15. Изложите порядок документирования результатов тестирования.

Примерные вопросы для защиты отчета по учебной практике

ПМ.02 Администрирование баз данных

(ОК 01; ОК 2; ОК 03; ОК 04; ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5)

1. Какие СУБД вы использовали на учебной практике?
2. Обоснуйте выбор СУБД для выполнения заданий практики.
3. Опишите процесс установки сервера баз данных.
4. Какие параметры конфигурации СУБД вы изменяли?
5. Как вы проверяли работоспособность СУБД после установки?
6. Какие графические клиенты для работы с СУБД вы использовали?
7. Опишите процесс создания базы данных.
8. Как создавались таблицы в вашей базе данных?
9. Какие типы данных вы использовали при создании таблиц?
10. Что такое первичный ключ? Как вы его определяли?
11. Что такое внешний ключ? Для каких связей вы его использовали?
12. Какие ограничения целостности (CHECK, NOT NULL, UNIQUE) вы применяли?
13. Как вы заполняли таблицы тестовыми данными?
14. Какие операции с данными (INSERT, UPDATE, DELETE) вы выполняли?
15. Опишите процесс импорта данных в базу данных.
16. Опишите процесс экспорта данных из базы данных.
17. Какие виды резервного копирования вы выполняли?
18. Какими утилитами вы пользовались для создания резервных копий?
19. Опишите процесс восстановления базы данных из резервной копии.
20. Как вы настраивали автоматическое резервное копирование?

Примерные вопросы для защиты отчета по учебной практике

ПМ.03 Обучение готовых моделей искусственного интеллекта

(ОК 01, ОК 2, ОК 03, ОК 04; ПК 3.1, ПК 3.2, ПК 3.3, ПК 3.4, ПК 3.5, ПК 3.6)

1. Как происходит сбор и предобработка данных из открытых источников.
2. Изложите порядок разработки программных модулей для анализа данных.
3. Как происходит разработка базовых моделей машинного обучения.
4. Как осуществляется визуализация результатов работы моделей ИИ.
5. Каков порядок написания юнит-тестов для ИИ-модулей
6. Как происходит установка и настройка среды разработки.
7. Опишите процесс разработки пользовательского интерфейса.

8. Как осуществляется интеграция предобученной модели в мобильное приложение.
9. Опишите порядок разработки прототипа мобильного приложения с ИИ.
10. Как осуществляется работа с системами контроля версий.
11. Опишите процесс разработки тестовых сценариев.
12. Каков порядок ручного тестирования ИИ-модулей.
13. Как осуществляется автоматизация тестирования.
14. Как происходит тестирование производительности.
15. Изложите порядок документирования результатов тестирования.

Примерные вопросы для защиты отчета по производственной практике (по профилю специальности)

ПМ.01 Разработка кода для обучения искусственного интеллекта
(ОК 01; ОК 2; ОК 04; ОК 05; ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5; ПК 1.6; ПК 1.7)

1. Как осуществляется сбор и обработка больших объемов данных для обучения моделей ИИ.
2. Опишите процесс Feature engineering и создания признаков.
3. Как осуществляется проектирование и реализация моделей машинного обучения.
4. Как осуществляется разработка моделей глубокого обучения.
5. Опишите порядок оптимизации моделей ИИ для повышения производительности
6. Как происходит валидация и тестирование моделей.
7. Опишите процесс разработки пайплайна машинного обучения.
8. Как осуществляется проектирование архитектуры мобильного ИИ-приложения.
9. Как происходит разработка сложных ИИ-приложений для мобильных платформ.
10. Как осуществляется интеграция ИИ-моделей в существующие информационные системы.
11. Опишите процесс оптимизации ИИ-моделей для мобильных устройств.
12. Как осуществляется разработка и публикация мобильных приложений.
13. Опишите процесс внедрения систем аналитики и мониторинга.

14. Как происходит обеспечение безопасности мобильного приложения.
15. Как происходит тестирование мобильного ИИ-приложения.
16. В чем суть разработки стратегии тестирования ИИ-систем.
17. Как происходит автоматизация тестирования программных продуктов.
18. Как осуществить проведение интеграционного тестирования сложных систем ИИ.
19. Как происходит тестирование качества ИИ-моделей.
20. Как осуществить нагрузочное и стресс-тестирование.
21. Как осуществляется процесс тестирования безопасности ИИ-приложений.
22. Как осуществить мониторинг производительности в реальных условиях.
23. Опишите процесс разработки и внедрения систем автоматизированного развертывания.
24. Изложите порядок документирования процесса отладки.

Примерные вопросы для защиты отчета по производственной практике (по профилю специальности)

ПМ.02 Администрирование баз данных

(ОК 01; ОК 2; ОК 03; ОК 04; ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5)

1. Как создавались пользователи в вашей базе данных?
2. Какие права доступа вы назначали пользователям?
3. Что такое роли в СУБД? Как вы их использовали?
4. Как проверить, какие права есть у конкретного пользователя?
5. Какие инструменты мониторинга производительности СУБД вы использовали?
6. Что такое slow query log? Как вы его настраивали?
7. Как вы анализировали план выполнения запроса с помощью EXPLAIN?
8. Что такое индексы в базах данных? Для чего они нужны?
9. Какие индексы вы создавали? На каких полях и почему?
10. Как вы оценивали эффективность созданных индексов?
11. Какой запрос вы оптимизировали? Что изменили?
12. Какие результаты дала оптимизация запросов?

13. Как вы настраивали удалённый доступ к серверу БД?
14. Какие файлы конфигурации изменяли для удалённого доступа?
15. Как проверить доступность сервера БД по сети?
16. Для какой предметной области вы проектировали базу данных?
17. Какие сущности вы выделили при проектировании?
18. Как вы строили ER-диаграмму? В каком инструменте?
19. До какой нормальной формы вы привели базу данных?
20. Какие типы JOIN вы использовали в своих запросах?
21. Какие агрегатные функции (SUM, COUNT, AVG) вы применяли?
22. Что такое подзапрос? Приведите пример из вашей работы.
23. Что такое хранимая процедура? Какую процедуру вы разработали?
24. Что такое триггер? Какой триггер вы создали?

Примерные вопросы для защиты отчета по производственной практике (по профилю специальности)

ПМ.03 Обучение готовых моделей искусственного интеллекта
(ОК 01, ОК 2, ОК 03, ОК 4; ПК 3.1, ПК 3.2, ПК 3.3, ПК 3.4, ПК 3.5, ПК 3.6)

1. Какие инструменты, библиотеки и фреймворки (например, TensorFlow, PyTorch, Scikit-learn) вы использовали в ходе практики? Обоснуйте их выбор.
2. Опишите общий пайплайн работы с моделью ИИ, который вы реализовали: от постановки задачи до финального результата.
3. С какими сложностями вы столкнулись в ходе практики и как их преодолевали?
4. Как вы подбирали датасет для обучения модели? Каковы его ключевые характеристики (объём, структура, типы данных)?
5. Какие методы предобработки данных вы применяли (очистка, нормализация, кодирование категориальных признаков и т.д.)? Приведите конкретные примеры.
6. Использовали ли вы аугментацию данных? Если да, какие техники и для какой задачи?
7. Как вы разделили данные на обучающую, валидационную и тестовую выборки? Обоснуйте выбранный подход.
8. Почему вы выбрали именно эту готовую модель ИИ для решения поставленной задачи? Сравните с альтернативными вариантами.
9. Какие гиперпараметры вы настраивали (скорость обучения, размер

- батча, количество эпох и т.д.)? Как выбирали их оптимальные значения?
10. Использовали ли техники регуляризации (Dropout, L1/L2-регуляризация) для предотвращения переобучения? Опишите их влияние на результат.
 11. Применяли ли трансферное обучение (transfer learning)? Если да, какую предобученную модель использовали и как её адаптировали?
 12. Опишите процесс обучения: динамика потерь (loss), точность (accuracy) и других метрик на обучающей и валидационной выборках.
 13. Какие метрики качества вы использовали для оценки модели (accuracy, precision, recall, F1-score, ROC-AUC и т.д.)? Почему именно их?
 14. Проанализируйте ошибки модели: какие типы ошибок она чаще всего совершает? Как это можно исправить?
 15. Проводили ли вы кросс-валидацию? Какие результаты она показала?
 16. Как вы интегрировали обученную модель в рабочую среду (например, через API, веб-приложение или скрипт)?
 17. Какие инструменты визуализации (Matplotlib, Seaborn, TensorBoard и т.д.) вы использовали для демонстрации результатов? Приведите примеры графиков или диаграмм.
 18. Приведите 2–3 реальных примера входных данных и соответствующих предсказаний модели. Насколько они соответствуют ожиданиям?
 19. Как можно использовать вашу модель в реальных бизнес-задачах? Приведите конкретный сценарий.
 20. Учитывали ли вы вопросы этичности и безопасности данных при обучении модели? Если да, какие меры предприняли?
 21. Соблюдались ли требования законодательства о защите персональных данных (например, 152-ФЗ) при работе с датасетом?
 22. Как вы оценивали потенциальную предвзятость (bias) модели? Были ли выявлены проблемы и как их решали?
 23. Каковы ключевые результаты вашей работы? Удалось ли достичь целевых показателей точности/эффективности?
 24. Какие улучшения можно внести в модель или процесс обучения для повышения качества предсказаний?

Критерии оценки защиты отчета по практике

Оценка «отлично» выставляется обучающемуся если структура отчета по практике соответствует рекомендуемой; все положения отчета

сформулированы правильно, применены корректные обозначения используемых в расчетах показателей, в результате анализа выполненных заданий, сделаны правильные выводы; в процессе защиты отчета последовательно, четко и логично изложены основные его положения и даны грамотные ответы на вопросы руководителя практики от филиала; в характеристике профессиональной деятельности обучающегося в период прохождения практики отмечена сформированность всех знаний и умений, предусмотренных программой практики.

Оценка «хорошо» выставляется обучающемуся если структура отчета соответствует рекомендуемой; обучающийся в процессе защиты отчета последовательно, достаточно четко изложил основные его положения, но допустил отдельные неточности в ответах на вопросы руководителя практики от филиала; в характеристике профессиональной деятельности обучающегося в период прохождения практики отмечена сформированность основных знаний, и умений, предусмотренных программой практики.

Оценка «удовлетворительно» выставляется обучающемуся если структура отчета не в полной мере соответствует рекомендуемой; обучающийся в процессе защиты отчета испытывает затруднения при ответах на вопросы руководителя практики от филиала, не способен ясно и четко изложить суть выполненных заданий и обосновать полученные результаты; в характеристике профессиональной деятельности обучающегося в период прохождения практики отмечена сформированность не менее 50% знаний, умений и навыков, предусмотренных программой практики.

Оценка «неудовлетворительно» выставляется обучающемуся если не подготовлен отчет по практике или структура отчета не соответствует рекомендуемой; в процессе защиты отчета обучающийся демонстрирует низкий уровень коммуникативности, неверно интерпретирует результаты выполненных заданий допускает принципиальные ошибки при ответе на вопросы и не способен к их исправлению без дополнительной подготовки; в характеристике профессиональной деятельности обучающегося в период прохождения практики отмечена несформированность знаний, умений и навыков, предусмотренных программой практики.

Экзаменационные вопросы и задания для устного (письменного) экзамена по модулю

Экзаменационные вопросы для устного (письменного) экзамена по модулю

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6, ПК 1.7)

1. Методы сбора и предобработки данных. Feature engineering и работа с несбалансированными данными.
2. Интеграция моделей машинного обучения в мобильные приложения с использованием TensorFlow Lite. Оптимизация моделей (квантизация, прунинг).
3. Метрики оценки качества моделей классификации: матрица ошибок (Confusion Matrix), Accuracy, Precision, Recall, F1-score.
4. Архитектуры нейронных сетей: сверточные (CNN) и рекуррентные (RNN) сети. Процессы обучения и функции активации.
5. Безопасность мобильных ИИ-приложений: защита моделей от кражи, безопасное хранение API-ключей, биометрическая аутентификация.
6. Спецификационные методы тест-дизайна: эквивалентное разбиение и анализ граничных значений.
7. Оценка качества моделей: кросс-валидация, регуляризация (L1/L2) и настройка гиперпараметров.
8. Развертывание мобильных приложений: CI/CD пайплайны, подготовка сборки (AAB/APK), публикация в магазинах приложений и пост-релизный мониторинг.
9. Автоматизация тестирования ИИ-систем: использование моков (mocks) и стабов, параметризация тестов, генерация синтетических данных.
10. Проектирование архитектуры ИИ-систем: микросервисы, контейнеризация (Docker, Kubernetes), REST API.
11. Разработка интерактивных мобильных ИИ-приложений: работа с камерой (CameraX), сенсорами устройства и дополненной реальностью (AR).
12. Тестирование ИИ-приложений: E2E-тестирование, нагрузочное тестирование и A/B тестирование.
13. Системы контроля версий (Git): ветвление, разрешение конфликтов, Pull Request, Code Review.
14. Платформы и инструменты мобильной разработки: структура Android-

- проекта, Gradle, основные UI-компоненты (RecyclerView, Navigation).
15. Планирование тестирования: тестовая стратегия, тестовый план, чек-листы и матрица трассировки требований.
 16. Алгоритмы машинного обучения: ансамблевые методы (случайный лес, градиентный бустинг), деревья решений.
 17. Гибридная архитектура мобильных ИИ-приложений: балансировка между локальным инференсом (Edge AI) и облачными вычислениями.
 18. Особенности тестирования ИИ-систем: ROC-кривая, AUC и анализ пороговых значений классификатора.
 19. Глубокое обучение: трансферное обучение (Transfer learning), аугментация данных, пакетная нормализация.
 20. Оптимизация потребления ресурсов мобильного устройства: профилирование памяти, CPU и батареи. Аппаратное ускорение (NNAPI, GPU Delegate).
 21. Управление дефектами: жизненный цикл дефекта, оценка серьезности (severity) и приоритета (priority). Регрессионное тестирование.
 22. Принципы «чистого кода», документирование, инструменты статического анализа кода.
 23. Жизненный цикл мобильного приложения: сбор и анализ обратной связи, интеграция аналитики, локализация.
 24. Тестирование API ИИ-сервисов: использование Postman/SoapUI, настройка тестовых стендов и заглушек.
 25. Введение в ИИ: обучение с подкреплением, системы рекомендаций, этические вызовы ИИ.
 26. Отладка мобильных приложений: использование Logcat, Android Profiler, Layout Inspector.
 27. Тестирование производительности: нагрузочное тестирование, стресс-тестирование, тестирование отказоустойчивости.
 28. Версионирование моделей и данных (MLflow, DVC). Мониторинг и надежность ИИ-систем.
 29. А/В тестирование функций мобильного приложения, проектирование UX/UI для ИИ-функций.
 30. Комплексное приемочное тестирование, оценка UX и интерпретируемости решений ИИ (XAI).
 31. Системы рекомендаций и поиск ассоциативных правил (алгоритм Apriori).
 32. Интеграция модели распознавания текста (OCR) и перевода текста в

- мобильное приложение.
- 33.Тестирование переходов состояний (State Transition Testing) и использование диаграмм причинно-следственных связей.
 - 34.Системы рекомендаций и поиск ассоциативных правил (алгоритм Apriori).
 - 35.Интеграция модели распознавания текста (OCR) и перевода текста в мобильное приложение.
 - 36.Тестирование переходов состояний (State Transition Testing) и использование диаграмм причинно-следственных связей.
 - 37.Автоэнкодеры для сжатия данных и методы аугментации данных в глубоком обучении.
 - 38.Оптимизация размера приложения (AAB/APK) и динамическая загрузка ИИ-моделей без перевыпуска приложения.
 - 39.Генераторы синтетических тестовых данных для ИИ и мок-тестирование внешних зависимостей (API).
 - 40.Валидация на временных рядах (Time Series Split) и анализ кривых обучения (Learning Curves).
 - 41.Мониторинг приложения после релиза (Crashlytics, ANR) и настройка автоматических хотфиксов.
 - 42.Оценка калибровки вероятностей и уверенности модели, валидация прогнозов классификатора.
 - 43.Наивный байесовский классификатор и его применение для текстовых данных (NLP).
 - 44.Обработка естественного языка в приложении и создание голосового помощника.
 - 45.Тестирование API ИИ-сервиса (Postman/SoapUI) и настройка тестового окружения с заглушками.
 - 46.Метод главных компонент (PCA) для уменьшения размерности данных.
 - 47.Аппаратное ускорение инференса (GPU Delegate, NNAPI) и профилирование производительности.
 - 48.Тестирование масштабируемости и нагрузки ИИ-системы, стресс-тестирование.
 - 49.Контейнеризация ИИ-систем с помощью Docker и развертывание в Kubernetes.
 - 50.Публикация в Google Play Console / Apple App Store и организация закрытого бета-тестирования.
 - 51.Комплексное приемочное тестирование ИИ-приложения перед релизом

- и оценка UX (интерпретируемость ХАІ).
- 52.Обеспечение безопасности и надежности ИИ-систем, стратегии резервного копирования.
 - 53.Биометрическая аутентификация и шифрование локальных данных приложения.
 - 54.Тестирование безопасности ИИ-приложений (Pentest) и тестирование отказоустойчивости.
 - 55.Интеграция очередей сообщений (RabbitMQ/Kafka) в архитектуру ИИ-системы.
 - 56.Создание приложения для умного дома с ИИ и взаимодействие с сенсорами устройства.
 - 57.Автоматизация регрессионного тестирования моделей и настройка генерации отчетов в CI/CD.
 - 58.Оптимизация ресурсов и управление затратами в облачной инфраструктуре для ИИ.
 - 59.Локализация приложения и создание системы уведомлений (архитектура каналов и приоритетов).
 - 60.Кросс-платформенное тестирование ИИ-интерфейсов и A/B тестирование в production.
 - 61.Разработка микросервисной архитектуры для ИИ-системы и проектирование REST API.
 - 62.Разработка приложения с дополненной реальностью (AR) и взаимодействие с камерой (CameraX).
 - 63.Тестирование на основе сценариев использования (Use Case) и оценка покрытия бизнес-логики.
 - 64.Калибровка вероятностей модели и анализ кривых Precision-Recall.
 - 65.Профилирование производительности ИИ-модели (латентность, память, батарея).
 - 66.Тестирование данных и пайплайнов предобработки, параметризованное тестирование.
 - 67.Версионирование моделей и данных с использованием MLflow и DVC.
 - 68.Работа в команде с Git (GitFlow, Code Review, Merge Requests) и написание технической документации.
 - 69.Разработка тестовых сценариев в соответствии с тестовым планом и шаблонами.
 - 70.LSTM сети для обработки последовательностей и анализ временных рядов.

- 71.Динамическая загрузка и обновление ИИ-моделей без перевыпуска приложения (OTA).
- 72.Тестирование пайплайнов данных и мониторинг дрейфа данных (Data Drift).
- 73.Этические вопросы и вызовы, связанные с развитием ИИ (предвзятость, приватность, безопасность).
- 74.Этические аспекты внедрения ИИ в мобильные приложения и защита персональных данных пользователей.
- 75.Роль тестировщика в команде разработки ИИ-продуктов и перспективы развития автоматизированного тестирования ИИ.

Экзаменационные задания для устного (письменного) экзамена по модулю

(ОК 01; ОК 02; ОК 04, ОК 05, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6, ПК 1.7)

1. Практическое задание (Кейс): Агротехническая компания разрабатывает мобильное приложение для распознавания болезней растений по фотографии листа. Приложение должно работать в полях без доступа к интернету. *Задание:* Опишите полный цикл решения задачи: как вы организуете сбор и предобработку датасета изображений, какие методы оптимизации модели выберете для её запуска на мобильном устройстве и как интегрируете её в Android-приложение, а также какие метрики качества будете использовать для оценки модели и как спроектируете тестовые сценарии для проверки работы приложения в условиях плохого освещения.
2. Практическое задание (Кейс-ситуация): Финтех-компания внедряет в мобильное банковское приложение функцию подтверждения крупных переводов с помощью распознавания лица (Face ID) и локальной ИИ-модели. *Задание:* Спроектируйте архитектуру нейронной сети для распознавания лиц. Опишите комплекс мер по защите биометрических данных и самого файла модели от реверс-инжиниринга на устройстве пользователя. Разработайте набор тестовых сценариев с использованием анализа граничных значений для проверки системы аутентификации (например, попытки входа при разном освещении, с маской, с фото вместо лица).
3. Практическое задание (Практико-ориентированное): Команда

разрабатывает мобильное приложение с голосовым помощником для управления умным домом. Обработка команд происходит как локально, так и через облачный API. *Задание:* Опишите процесс валидации и настройки гиперпараметров NLP-модели. Спроектируйте CI/CD пайплайн для автоматической сборки, тестирования и публикации обновления приложения. Опишите, как вы будете писать юнит-тесты для модуля обработки команд, используя моки для изоляции от внешнего облачного API, и как настроите генерацию отчетов в пайплайне.

4. Практическое задание (Кейс): Музей заказывает разработку мобильного интерактивного гида. Приложение должно в реальном времени распознавать экспонаты через камеру, накладывать AR-метки и генерировать адаптивный аудиорассказ. *Задание:* Спроектируйте серверную микросервисную архитектуру для хранения данных об экспонатах и генерации речи. Опишите техническое решение для передачи видеопотока с камеры в ИИ-модель без блокировки UI на мобильном устройстве. Разработайте план E2E-тестирования пользовательского сценария «Посещение выставки» и предложите метрики для A/B тестирования двух разных алгоритмов рекомендаций маршрута.
5. Практическое задание (Ситуационное): Стартап разрабатывает MVP мобильного приложения для трекинга физической активности с ИИ-анализом техники выполнения упражнений по видео. Команда состоит из трех разработчиков. *Задание:* Опишите регламент командной работы в Git (модель ветвления, Code Review) для параллельной разработки бэкенда и мобильного клиента. Спроектируйте архитектуру UI мобильного приложения для отображения ленты тренировок и результатов анализа. Разработайте тестовый план и чек-лист для смоук-тестирования релизной версии приложения, включая матрицу трассировки бизнес-требований.
6. Практическое задание (Кейс-ситуация): Разрабатывается мобильное приложение для логистической компании, которое прогнозирует время доставки и строит маршруты с учетом пробок и погоды. *Задание:* Обоснуйте выбор алгоритма машинного обучения для прогнозирования временных рядов и оценки важности признаков. Спроектируйте гибридную архитектуру приложения: какие данные обрабатываются локально, а какие отправляются в облако, и как приложение ведет себя при потере сети. Опишите, как вы будете использовать ROC-кривую и AUC для выбора оптимального порога уверенности модели, при

котором система будет переключаться с локального прогноза на облачный.

7. Практическое задание (Практико-ориентированное): Компания разрабатывает приложение для бортового компьютера дрона, которое в реальном времени обнаруживает объекты на местности. Датасет небольшой, а ресурсы дрона сильно ограничены. *Задание:* Опишите, как вы примените трансферное обучение и аугментацию данных для повышения качества модели на малой выборке. Какие методы оптимизации и аппаратного ускорения вы используете, чтобы модель работала быстро и не разряжала батарею дрона за 5 минут? После релиза пользователи сообщили о редких «вылетах» приложения при обнаружении специфических объектов. Опишите процесс управления этим дефектом и план регрессионного тестирования перед выпуском патча.
8. Практическое задание (Кейс): Разрабатывается мобильный агрегатор новостей с персонализированной ИИ-лентой рекомендаций. *Задание:* Опишите стандарты оформления и документирования кода для серверного модуля рекомендаций, а также инструменты статического анализа. Как вы встроите аналитику в мобильное приложение для сбора фидбека (лайки, время чтения) и как организуете локализацию интерфейса? Разработайте коллекцию запросов в Postman для тестирования REST API рекомендательного сервиса, включая проверку обработки некорректных JSON-запросов и таймаутов.
9. Практическое задание (Ситуационное): Мобильный банк запускает ИИ-чат-бота для автоматической поддержки клиентов в приложении. *Задание:* Опишите концепцию обучения чат-бота с подкреплением для улучшения качества ответов и этические аспекты его внедрения. Пользователи жалуются, что интерфейс чата «фризит» при загрузке истории переписки. Опишите алгоритм поиска причины с помощью Android Profiler и Logcat. Разработайте сценарий нагрузочного и стресс-тестирования серверной части чат-бота для проверки его устойчивости при одновременном подключении 100 000 пользователей.
10. Практическое задание (Кейс-ситуация): Разрабатывается мобильное приложение для подбора индивидуального плана питания и тренировок с использованием ИИ. *Задание:* Опишите процесс версионирования датасетов пользователей и версий модели рекомендаций с помощью DVC/MLflow. Спроектируйте UX интерфейса, который объясняет пользователю, почему ИИ рекомендовал именно этот рацион, и опишите дизайн A/B теста для проверки двух вариантов отображения этих

объяснений. Составьте чек-лист комплексного приемочного тестирования приложения перед релизом, уделив особое внимание проверке интерпретируемости и понятности ИИ-рекомендаций для конечного пользователя.

11. Практическое задание (Кейс): Разрабатывается мобильное приложение-переводчик для туристов, которое позволяет навести камеру на вывеску на иностранном языке, распознать текст и отобразить перевод в дополненной реальности. *Задание:* Опишите, как вы будете собирать и размечать датасет для OCR. Спроектируйте архитектуру Android-приложения с использованием CameraX и TFLite для обработки видеопотока без лагов. Разработайте тестовые сценарии на основе диаграммы переходов состояний (например, состояния: «Нет сети», «Камера занята», «Текст не распознан», «Перевод получен») для проверки устойчивости приложения.
12. Практическое задание (Ситуационное): Мобильное фитнес-приложение использует ИИ для прогнозирования количества сожженных калорий и усталости пользователя в реальном времени на основе данных с акселерометра и гироскопа, адаптируя сложность тренировки. *Задание:* Обоснуйте применение обучения с подкреплением для персонализации плана тренировок. Опишите процесс сбора данных с сенсоров и реализации фоновой отправки умных push-уведомлений. Выберите и рассчитайте метрики регрессии для оценки точности прогноза калорий, а также предложите метрики для кластеризации пользователей по уровням активности.
13. Практическое задание (Практико-ориентированное): Команда разрабатывает мобильный фоторедактор с ИИ-функцией «улучшение качества старых снимков». Модель весит более 300 МБ, что превышает лимиты магазинов приложений. *Задание:* Опишите, как использовать автоэнкодеры для сжатия и восстановления изображений, и как применить аугментацию для расширения обучающей выборки. Предложите архитектуру доставки модели: как уменьшить базовый APK и настроить динамическую подгрузку весов модели по Wi-Fi. Как вы будете генерировать синтетические тестовые данные (искаженные фото) и использовать моки для тестирования процесса загрузки модели с сервера.
14. Практическое задание (Кейс): Разрабатывается ИИ-система для предиктивного обслуживания промышленного оборудования (вибрационные датчики), данные с которых поступают в виде временных рядов. Мобильное приложение используется инженерами

для получения алертов. Задание: Опишите процесс валидации модели на временных рядах и интерпретацию кривых обучения для выявления переобучения. Как вы организуете мониторинг мобильного приложения инженеров на предмет ANR и крашей при получении алертов. Объясните, зачем нужна калибровка вероятностей модели (чтобы алерт срабатывал только при 99% уверенности) и как протестировать этот порог.

15. Практическое задание (Ситуационное): В корпоративном мобильном приложении внедряется голосовой помощник, который должен классифицировать голосовые команды сотрудников (например, «создать заявку», «отправить отчет») и маршрутизировать их. Задание: Обоснуйте использование наивного байесовского классификатора для быстрого NLP-интента на сервере. Опишите интеграцию механизма распознавания речи (Speech-to-Text) и синтеза ответа в мобильном клиенте. Составьте коллекцию тестов в Postman для API классификации интентов, включая тесты с моками (заглушками) базы данных пользователей.
16. Практическое задание (Практико-ориентированное): Мобильное приложение для экологов должно анализировать спектральные данные с портативного датчика качества воды, передавая их на сервер для классификации загрязнений. Признаков очень много, а канал связи в полевых условиях нестабилен. Задание: Опишите применение PCA для сокращения количества передаваемых признаков без потери критической информации. Как вы будете использовать NNAPI на устройстве для ускорения работы модели фильтрации шумов. Спроектируйте сценарий нагрузочного и стресс-тестирования серверной части, которая принимает данные от 10 000 датчиков одновременно.
17. Практическое задание (Практико-ориентированное): Мобильное приложение для экологов должно анализировать спектральные данные с портативного датчика качества воды, передавая их на сервер для классификации загрязнений. Признаков очень много, а канал связи в полевых условиях нестабилен. Задание: Опишите применение PCA для сокращения количества передаваемых признаков без потери критической информации. Как вы будете использовать NNAPI на устройстве для ускорения работы модели фильтрации шумов. Спроектируйте сценарий нагрузочного и стресс-тестирования серверной части, которая принимает данные от 10 000 датчиков одновременно.
18. Практическое задание (Ситуационное): Разрабатывается мобильный

банк с функцией ИИ-скоринга кредитов прямо в приложении. Приложение хранит локальные токены и использует FaceID для входа. Задание: Какие стратегии резервного копирования и аварийного восстановления вы заложите в архитектуру сервера ИИ-скоринга. Опишите реализацию биометрической аутентификации и шифрования локальной БД (Room/SQLite) на устройстве. Как вы будете проводить тестирование на проникновение (попытка извлечения модели скоринга из APK) и тестирование отказоустойчивости при обрыве связи с сервером в момент принятия решения.

19. Практическое задание (Практико-ориентированное): Система «Умный дом» анализирует показания датчиков (температура, влажность, движение) и предсказывает оптимальный режим отопления. Обработка данных должна быть асинхронной. Задание: Обоснуйте использование Kafka/RabbitMQ для передачи потоков данных с датчиков к ИИ-модели. Опишите архитектуру мобильного клиента, который агрегирует данные с BLE-сенсоров и отправляет их в очередь. Напишите псевдокод настройки CI/CD пайплайна, который при каждом изменении весов модели автоматически запускает регрессионные тесты и формирует отчет о падении метрик.
20. Практическое задание (Кейс): Международная компания выпускает мобильное приложение с ИИ-диагностикой кожных заболеваний. Серверная часть требует мощных GPU. Задание: Какие методы оптимизации облачных ресурсов (автоскейлинг, spot-инстансы) вы примените для снижения затрат на обучение и инференс моделей. Как вы организуете локализацию UI и настройку пуш-уведомлений (напоминания о приеме лекарств) с учетом разных часовых поясов и приоритетов. Спроектируйте A/B тест для проверки двух разных алгоритмов ранжирования рекомендаций врачей и опишите метрики кросс-платформенного тестирования (Android vs iOS).
21. Практическое задание (Ситуационное): Мобильный AR-гид по музею. Наводя камеру на картину, пользователь видит поверх нее биографию художника, сгенерированную ИИ. Задание: Спроектируйте REST API для микросервиса, который принимает координаты AR-маркера и возвращает сгенерированный текст. Опишите процесс интеграции ARCore и CameraX для привязки ИИ-текста к физическому объекту в реальном времени. Разработайте 3 детальных Use Case (сценария использования) и составьте на их основе тестовые сценарии, оценив процент покрытия бизнес-требований.
22. Практическое задание (Ситуационное): Мобильный AR-гид по музею.

Наводя камеру на картину, пользователь видит поверх нее биографию художника, сгенерированную ИИ. Задание: Спроектируйте REST API для микросервиса, который принимает координаты AR-маркера и возвращает сгенерированный текст. Опишите процесс интеграции ARCore и CameraX для привязки ИИ-текста к физическому объекту в реальном времени. Разработайте 3 детальных Use Case (сценария использования) и составьте на их основе тестовые сценарии, оценив процент покрытия бизнес-требований.

23. Практическое задание (Кейс): В компанию пришла новая команда Data Science и Mobile Developers. Необходимо настроить с нуля процесс совместной работы над ИИ-продуктом. Задание: Опишите регламент версионирования датасетов и артефактов моделей с помощью DVC и отслеживания экспериментов в MLflow. Предложите модель ветвления в Git (GitFlow), регламент проведения Code Review и структуру технической документации для передачи модулей в поддержку. Создайте шаблон тестового сценария (Test Case) и напишите по нему 2 тест-кейса для проверки интеграции новой версии модели в мобильное приложение.
24. Практическое задание (Ситуационное): Носимое устройство (умные часы) с companion-приложением использует ИИ для предсказания гипогликемии у диабетиков на основе непрерывного мониторинга глюкозы (временные ряды). Задание: Обоснуйте выбор архитектуры LSTM для прогнозирования временных рядов глюкозы. Опишите механизм OTA-обновления весов модели LSTM прямо в приложении, когда сервер обнаружил новый паттерн. Как вы будете тестировать пайплайн сбора данных и настраивать алерты для QA-команды в случае обнаружения Data Drift (изменения распределения данных у пользователей).
25. Практическое задание (Кейс): HR-агентство внедряет мобильное приложение и веб-портал для автоматического скрининга резюме кандидатов с помощью ИИ. В обществе возникают опасения относительно дискриминации по полу и возрасту. Задание: Опишите методы выявления и устранения предвзятости (bias) в датасете резюме на этапе предобработки. Какие этические нормы и механизмы согласия (GDPR/152-ФЗ) вы заложите в UI/UX мобильного приложения при сборе персональных данных кандидатов. Сформулируйте, какую роль должен играть QA-инженер в создании тестов на «справедливость» (Fairness Testing) модели, и какие перспективы имеет автоматизация таких проверок.

Критерии оценки устного (письменного) экзамена по модулю

Оценка «отлично» выставляется, если ответ студента полный, логичный, структурированный и исчерпывающий. Студент демонстрирует глубокое понимание теоретического материала, свободно и корректно оперирует профессиональной терминологией. Практическое задание выполнено в полном объеме. Студент предлагает комплексное, архитектурно грамотное решение, которое полностью соответствует поставленной задаче. Студент уверенно и самостоятельно аргументирует выбор методов, инструментов и проектных решений, не допускает фактических ошибок и способен дать четкие ответы на дополнительные вопросы преподавателя.

Оценка «хорошо» выставляется, если ответ студента достаточно полный и логичный, демонстрирует уверенное владение теоретическим материалом. Допускаются незначительные шероховатости в формулировках или мелкие неточности в терминологии. Практическое задание выполнено, предложенное решение в целом верное и охватывает основные аспекты задачи, но содержит незначительные недочеты в аргументации или выборе оптимальных инструментов. Студент способен самостоятельно исправить допущенные ошибки или дополнить ответ после минимального наводящего вопроса преподавателя.

Оценка «удовлетворительно» выставляется, если ответ студента фрагментарный, демонстрирует поверхностное понимание теоретического материала и неуверенное владение базовой терминологией. Логика изложения нарушена. Практическое задание выполнено с существенными недочетами: предложенное решение охватывает лишь часть задачи, студент испытывает трудности с обоснованием выбора методов и не видит взаимосвязи между этапами работы. Для формирования полного ответа и принятия верных проектных решений студенту требуются существенные наводящие вопросы и помощь преподавателя.

Оценка «неудовлетворительно» выставляется, если ответ студента не соответствует поставленным вопросам, демонстрирует отсутствие понимания теоретического материала и неумение применять его на практике. Студент не владеет базовой профессиональной терминологией. Практическое задание не выполнено или содержит грубые ошибки. Предложенное решение не соответствует техническому заданию, профессиональным стандартам и логике предметной области. Студент не может аргументировать свои действия, не понимает специфику задачи и не способен ответить на дополнительные вопросы преподавателя.