

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)**

**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ
ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ
(АКТ (ф) СПбГУТ)**

УТВЕРЖДАЮ

И.о. зам. директора по учебной работе

К.А. Семенцына

2025 г.

КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ

по профессиональному модулю

ПМ.01 РАЗРАБОТКА КОДА ДЛЯ ОБУЧЕНИЯ

ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

программы подготовки специалистов среднего звена

по специальности СПО

09.02.13 Интеграция решений с применением технологий

искусственного интеллекта

г. Архангельск

2025

Организация-разработчик: АКТ (ф) СПбГУТ.

Разработчики:

М.Н. Нехлебаева, преподаватель высшей квалификационной категории
АКТ (ф) СПбГУТ.

С.В. Лукина, преподаватель высшей квалификационной категории АКТ
(ф) СПбГУТ.

Рассмотрено и одобрено цикловой комиссией Информационных
технологий и математических дисциплин

Протокол № 3 от 12 ноября 2025 г.

Председатель Нехлеб М.Н. Нехлебаева

КОМПЛЕКТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ
для промежуточной аттестации (экзамен квалифицированный)
по ПМ.01. Разработка кода для обучения искусственного интеллекта
для специальности

09.02.13 Интеграция решений с применением технологий искусственного интеллекта

Экзамен квалификационный является итоговой формой контроля, подводит итог освоения ПМ.01. Разработка кода для обучения искусственного интеллекта. К сдаче зачета допускаются студенты, успешно выполнившие все практические и лабораторные работы.

Всем студентам предлагаются варианты подобных заданий, с целью обеспечения равных условий выполнения. При выполнении заданий студенты могут пользоваться персональными компьютерами и наглядными пособиями, материалами справочного характера, нормативными документами и различными образцами, которые разрешены к использованию на экзамене квалификационном и указаны в билете в разделе «Инструкция».

Результаты экзамена квалификационного определяются на основании оценочной ведомости и/или результатов решения профессиональных задач оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно», вносятся в итоговую ведомость экзамена квалификационного аттестационной комиссии и объявляются в тот же день.

Решение аттестационной комиссии об итоговой оценке студента по экзамену квалификационному принимается на закрытом заседании простым большинством голосов членов аттестационной комиссии, участвующих в заседании. При равном числе голосов - голос председателя является решающим.

Время проведения экзамена – 6 часов.

В результате освоения ПМ.01. студент должен освоить следующие профессиональные компетенции:

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 контроля версий программного кода.

ПК 1.5. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.6. Выполнять тестирование программного кода.

ПК 1.7. Составлять тестовые сценарии.

В ходе проведения экзамена проверяется сформированность следующих навыков:

- разработка программные модули для систем искусственного интеллекта;
- выполнение предобработку и анализ данных;
- обучение, оценивание и улучшение модели машинного и глубокого обучения;
- проектирование и развертывание масштабируемых ИИ-систем;
- знание основных направлений, истории и этических аспектов ИИ;
- методы сбора, предобработки и анализа данных;
- алгоритмы машинного обучения и глубокого обучения;
- методы оценки и валидации моделей;
- принципы проектирования и контейнеризации ИИ-систем.

Шкала оценивания образовательных результатов:

№	Тип задания	Критерии оценки
1	Устные ответы, письменные развернутые ответы	Оценка «5» ставится в том случае, если обучающийся правильно понимает суть вопроса, дает точное определение и истолкование основных понятий; правильно анализирует условие задачи (вопроса), ответ логичен, умеет выстроить алгоритм поиска ответа самостоятельно; строит ответ по собственному плану, сопровождает ответ новыми примерами, умеет применить знания в новой ситуации; может установить связь между изучаемым и ранее изученным материалом из курса дисциплины, а также с материалом, усвоенным при изучении других дисциплин/модулей. Оценка «4» ставится, если ответ обучающегося удовлетворяет основным требованиям к ответу на оценку 5, но дан без использования собственного плана, новых примеров, без применения знаний в новой ситуации, без использования связей с ранее изученным материалом и материалом, усвоенным при изучении других дисциплин/модулей; обучающийся допустил одну ошибку или не более двух недочетов и может их исправить самостоятельно или с небольшой помощью преподавателя. Оценка «3» ставится, если обучающийся правильно понимает суть вопроса, но в ответе имеются отдельные пробелы в усвоении вопросов курса дисциплины, не препятствующие дальнейшему усвоению программного материала; умеет применять полученные знания при решении простых задач (заданий, вопросов) по готовому алгоритму; допустил не более одной грубой ошибки и двух недочетов, не более одной грубой и одной негрубой ошибки, не более двух-трех негрубых ошибок, одной негрубой ошибки и трех недочетов; допустил четыре-пять недочетов. Оценка «2» ставится, если обучающийся не овладел основными знаниями и умениями в соответствии с требованиями программы и допустил больше ошибок и недочетов, чем необходимо для оценки.
2	Тесты	«5» - 100 – 91% правильных ответов

		«4» - 90 - 70% правильных ответов «3» - 69 – 52% правильных ответов «2» - 51% и менее правильных ответов
3	Практические задания	Оценка «5» выставляется, если обучающийся активно работает в течение всего практического занятия, правильно выполняет все этапы практического задания, на 100% выполняет практическую работу, заполняет основную надпись на чертеже. Оценка «4» выставляется при условии соблюдения следующих требований: обучающийся активно работает в течение практического занятия, правильно выполняет все этапы практического задания, на 85% выполняет практическую работу, допущены неточности, некоторые незначительные ошибки при выполнении практической работы, заполняет основную надпись на чертеже. Оценка «3» выставляется в том случае, когда обучающийся в целом овладел сути вопросов по данной теме, но на занятии 20 ведет себя пассивно, допускает грубые ошибки при выполнении практической работы, отвечает на теоретические вопросы по данной теме, но не может обобщить и сделать четкие логические выводы, работа выполнена на 60%, заполняет основную надпись на чертеже. Оценка «2» выставляется в случае, когда обучающийся обнаружил несостоятельность осветить вопросы по данной теме практической работы или вопросы освещены неправильно. Практическая работа выполнена с грубыми ошибками, отсутствуют понимания основной сути практической работы, выводы, обобщения, обнаружено неумение оформлять техническую документацию в соответствии с действующей нормативной базой.

Типовые задания для оценки освоения МДК 01.01. Разработка программных модулей в системах искусственного интеллекта

Теоретические вопросы

1. Дайте определение машинного обучения как подраздела искусственного интеллекта.
2. Что такое обучение с учителем и приведите пример алгоритма для него?
3. Что такое обучение без учителя и приведите пример алгоритма для него?
4. В чем основное различие между задачами классификации и регрессии?
5. Опишите суть обучения с подкреплением и приведите пример его применения.
6. Что такое "искусственный интеллект" в широком смысле этого слова?
7. Назовите ключевые факторы, способствовавшие современному буму в области ИИ.
8. Что такое "рекомендательная система" и на каких принципах она строится?
9. Какова роль больших данных (Big Data) в развитии машинного обучения?
10. Что такое "разметка данных" и почему она важна для обучения с учителем?
11. Для чего необходима предобработка данных перед построением модели?
12. Что такое "пропущенные значения" и назовите три способа работы с ними.
13. Дайте определение кодирования категориальных признаков. Зачем оно нужно?
14. В чем разница между кодированием меток (Label Encoding) и One-Hot кодированием?

15. Что такое "выбросы" в данных и как они могут повлиять на модель?
16. Объясните, что такое нормализация данных и какую проблему она решает.
17. Объясните, что такое стандартизация данных и какую проблему она решает.
18. Что такое "несбалансированные данные" и каковы методы борьбы с этим?
19. Как веб-скрапинг используется в конвейере сбора данных для ИИ?
20. Что такое SQL и как он применяется для подготовки данных?
21. Опишите основной принцип работы алгоритма k-ближайших соседей (k-NN).
22. Как работает алгоритм "Дерево решений" для задачи классификации?
23. В чем заключается основная идея метода опорных векторов (SVM)?
24. Что такое "линейная регрессия" и как оцениваются ее параметры?
25. Для каких задач используется логистическая регрессия, несмотря на название "регрессия"?
26. Опишите алгоритм кластеризации k-средних (k-means).
27. Что такое "центроид" в контексте алгоритма k-means?
28. Какова цель иерархической кластеризации?
29. Что такое "ансамбли моделей" и какую проблему они решают?
30. Опишите принцип работы алгоритма случайного леса (Random Forest).
31. Что такое "переобучение" модели машинного обучения?
32. Что такое "недообучение" модели машинного обучения?
33. Почему важно разделять данные на обучающую и тестовую выборки?
34. Что такое кросс-валидация и зачем она применяется?
35. Дайте определение метрики "Точность" (Accuracy) и опишите ее ограничения.
36. Дайте определение метрики "Precision" и напишите ее формулу.
37. Дайте определение метрики "Recall" и напишите ее формулу.
38. Что такое F1-мера и почему она часто полезнее точности?
39. Что такое матрица ошибок и какую информацию она предоставляет?
40. Что показывает ROC-кривая?
41. Что такое AUC-ROC и как интерпретируется его значение?
42. Как регуляризация помогает бороться с переобучением?
43. В чем разница между регуляризацией L1 (Lasso) и L2 (Ridge)?
44. Что такое отбор признаков и как он связан с регуляризацией L1?
45. Что такое гиперпараметры модели и чем они отличаются от обычных параметров?
46. Из каких основных компонентов состоит искусственный нейрон?
47. Что такое функция активации и какую роль она играет в нейронной сети?
48. Сравните функции активации Sigmoid и ReLU. Каковы их преимущества и недостатки?
49. Что такое функция потерь и какова ее роль в обучении нейронной сети?
50. Опишите архитектуру многослойного перцептрона.
51. В чем заключаются ключевые особенности сверточных нейронных сетей (CNN)?
52. Для каких задач преимущественно используются рекуррентные нейронные сети (RNN)?
53. Что такое механизм внимания в нейронных сетях?
54. Что такое архитектура Transformer и в чем ее основное преимущество перед RNN?
55. Опишите процесс обратного распространения ошибки.
56. Как работает оптимизатор Adam?
57. Что такое пакетная нормализация и для чего она применяется?
58. Что такое "заморозка" слоев нейронной сети и когда это используется?
59. Что такое передача обучения и каковы ее преимущества?
60. Опишите проблему "исчезающего градиента" в глубоких сетях.

Практические задания

1. Составьте список из 5 потенциальных проблем с приватностью, которые могут возникнуть при использовании умных домашних ассистентов.
2. Используя библиотеку requests в Python, получите данные о курсе криптовалют с API CoinGecko.
3. Напишите скрипт на Python с использованием BeautifulSoup для извлечения всех цен товаров с страницы интернет-магазина.
4. Загрузите датасет с пропущенными значениями и выведите процент пропусков в каждом столбце.
5. Для датасета из предыдущего задания удалите строки, где количество пропусков превышает 50%.
6. Для категориального столбца "Профессия" примените Label Encoding, используя LabelEncoder из sklearn.
7. Нормализуйте числовой столбец "Доход" с помощью Min-Max Scaling, используя MinMaxScaler.
8. Стандартизируйте числовой столбец "Рост" с помощью StandardScaler.
9. Напишите функцию для обнаружения выбросов в столбце с использованием z-score метода.
10. Разделите датасет на обучающую (80%) и тестовую (20%) выборки, стратифицируя по целевому признаку.
11. Обучите модель линейной регрессии на данных о стаже работы и зарплате. Сделайте прогноз зарплаты для сотрудника с 5-летним стажем.
12. Создайте модель логистической регрессии для классификации emails на "спам/не спам".
13. Реализуйте алгоритм k-ближайших соседей (kNN) с $k=5$ для классификации типов вина.
14. Обучите модель случайного леса с 100 деревьями для классификации ирисов Фишера.
15. Примените алгоритм DBSCAN для кластеризации данных и выберите оптимальные параметры eps и min_samples.
16. Сравните результаты кластеризации k-means и агломеративной кластеризации на синтетическом датасете.
17. Для модели бинарной классификации рассчитайте метрики Precision, Recall и F1-score для каждого класса.
18. Постройте и визуализируйте матрицу ошибок с помощью seaborn.heatmap.
19. Рассчитайте ROC-AUC score для модели и определите оптимальный порог классификации.
20. Постройте precision-recall кривую для несбалансированного датасета.
21. Проведите кросс-валидацию на 10 фолдах для модели градиентного бустинга.
22. Добавьте L1-регуляризацию в модель линейной регрессии и проанализируйте, какие признаки были отсеяны.
23. Настройте гиперпараметр n_estimators для случайного леса, используя RandomizedSearchCV.
24. Создайте архитектуру нейронной сети с двумя скрытыми слоями (64 и 32 нейрона) для многоклассовой классификации.
25. Скомпилируйте модель, выбрав оптимизатор 'adam', функцию потерь 'categorical_crossentropy' и метрику 'accuracy'.
26. Обучите модель на датасете fashion-MNIST и добейтесь точности выше 85% на валидации.
27. Создайте сверточную нейронную сеть с двумя сверточными слоями и слоем dropout для классификации изображений CIFAR-10.
28. Обучите CNN на CIFAR-10 в течение 50 эпох с использованием аугментации данных.
29. Создайте RNN с двумя LSTM слоями для прогнозирования цен акций.

30. Измените оптимизатор с 'adam' на 'sgd' и сравните скорость сходимости модели.
31. Добавьте Batch Normalization после каждого сверточного слоя в CNN.
32. Спроектируйте модульную систему для предсказания оттока клиентов банка.
33. Напишите FastAPI приложение, которое принимает JSON с features и возвращает prediction модели.
34. Создайте Dockerfile, который копирует модель, зависимости и запускает FastAPI приложение.
35. Соберите Docker-образ с тегом "ml-api:v1.0".
36. Запустите контейнер с пробросом порта 8000 и протестируйте эндпоинт /predict.
37. Напишите docker-compose.yml для развертывания приложения с MongoDB для хранения предсказаний.
38. Создайте Kubernetes Deployment с 2 репликами и ресурсными ограничениями: 512MB RAM, 0.5 CPU.
39. Настройте Horizontal Pod Autoscaler для автоматического масштабирования от 2 до 10 подов при нагрузке >80% CPU.
40. Реализуйте мониторинг модели с помощью Prometheus метрик (количество запросов, время инференса, ошибки).

Типовые задания для оценки освоения МДК 01.02. Разработка мобильных приложений с поддержкой искусственного интеллекта

Теоретические вопросы

1. Определение качества программного обеспечения согласно ГОСТ Р ИСО/МЭК 25051.
2. Основные метрики качества программного обеспечения.
3. Уровни тестирования программных приложений.
4. Ключевые отличия модульного, интеграционного, системного и приемочного тестирования.
5. Типы тестирования, основанные на спецификациях.
6. Сущность тестирования на основе сценариев использования.
7. Основные виды тестирования производительности.
8. Цели и задачи регрессионного тестирования.
9. Основные этапы жизненного цикла дефекта.
10. Уровни серьезности дефектов в программном обеспечении.
11. Процесс определения целей тестирования для различных уровней.
12. Назначение и структура тестового сценария.
13. Основные разделы тестового плана.
14. Отличие чек-листа от тестового сценария.
15. Понятие тестового пакета и задания на тестирование.
16. Основные инструменты проектирования тестов.
17. Сущность техники тест-дизайна «Эквивалентное разбиение».
18. Принцип анализа граничных значений.
19. Техника тестирования с использованием таблиц решений.
20. Процесс тестирования переходов состояний системы.
21. Основные метрики для оценки качества классификационных моделей.
22. Компоненты матрицы ошибок (Confusion Matrix): TP, FP, FN, TN.
23. Формулы и интерпретация метрик Precision (Точность) и Recall (Полнота).
24. Недостатки метрики Accuracy (Точность) в несбалансированных выборках.
25. Смысл и расчет F1-score (F-мера).
26. Назначение и построение ROC-кривой (Receiver Operating Characteristic curve).
27. Интерпретация площади под ROC-кривой (AUC - Area Under Curve).

28. Ограничения применения ROC-кривой и AUC.
29. Основные метрики для оценки моделей регрессии.
30. Формула и интерпретация MAE (Mean Absolute Error).
31. Формула и интерпретация MAPE (Mean Absolute Percentage Error).
32. Понятие оптимальной сложности модели и переобучения.
33. Основные метрики для оценки качества кластеризации.
34. Процесс обучения модели линейной регрессии.
35. Инструменты для автоматизации выполнения тестовых примеров.
36. Принципы интеграции автоматизированных тестов в процесс CI/CD.
37. Назначение и типы генераторов тестовых данных.
38. Основные проблемы тестирования мобильных ИИ-приложений.
39. Понятие тестового окружения (стенда) и его компоненты.
40. Назначение и виды заглушек (mocks, stubs) в тестировании.
41. Отличия E2E (End-to-End) тестов от модульных и интеграционных.
42. Цели и методы тестирования отказоустойчивости систем.
43. Цели и методы стресс-тестирования.
44. Основные аспекты тестирования безопасности ИИ-приложений.
45. Инструменты автоматизации тестирования веб-API (например, Postman, SOAPUI).
46. Методы мониторинга производительности ИИ-приложений.
47. Процесс профилирования ресурсов (CPU, память) ИИ-приложения.
48. Цели и этапы юзабилити-тестирования интерфейса.
49. Методы тестирования совместимости приложения с различными браузерами.
50. Процесс тестирования API веб-сервисов.
51. Принципы мониторинга ИИ-моделей в производственной среде.
52. Системы для визуализации метрик производительности и данных мониторинга.
53. Понятие «сервиса» в Android для выполнения длительных фоновых задач.
54. Основные принципы Material Design.
55. Роль файла AndroidManifest.xml в проекте Android.
56. Назначение и основные команды системы контроля версий Git (commit, push, pull).
57. Отличия GitLab от GitHub.
58. Цели и преимущества автоматизированного тестирования.
59. Назначение фреймворка Espresso для тестирования UI в Android.
60. Основные этапы подготовки Android-приложения к публикации в Play Market.

Практические задания

1. Определите цели тестирования для модульного, интеграционного и системного уровней для заданного функционального требования.
2. Разработайте набор юнит-тестов для функции предварительной обработки данных (нормализации числовых признаков).
3. Реализуйте оценку качества бинарного классификатора с построением ROC-кривой и расчетом AUC-ROC.
4. Рассчитайте метрики Precision, Recall и F1-score на основе предоставленной матрицы ошибок.
5. Проведите интеграционное тестирование пайплайна машинного обучения, включающего этапы загрузки данных, предобработки и предсказания модели.
6. Подготовьте тестовый пакет для функционального тестирования модуля классификации текстов.
7. Разработайте тестовый сценарий для проверки функции регистрации пользователя в веб-приложении.

8. Протестируйте веб-приложение с интегрированной моделью ИИ на отказоустойчивость при недоступности сервера предсказаний.
9. Проведите нагрузочное тестирование REST API, используемого для получения предсказаний от модели.
10. Проанализируйте требования к системе и примените технику эквивалентного разбиения для создания тестовых случаев.
11. Используйте анализ граничных значений для проектирования тестов для поля ввода, принимающего значения от 1 до 100.
12. Создайте и примените таблицу решений для тестирования логики принятия решения о скидке на основе условий.
13. Постройте диаграмму причинно-следственных связей для тестирования функции оформления заказа.
14. Реализуйте тестирование на основе сценария использования «Поиск и бронирование отеля».
15. Настройте окружение для тестирования с использованием заглушек (mocks) для внешнего платежного шлюза.
16. Напишите юнит-тесты для функции вычисления метрики MAPE (Mean Absolute Percentage Error).
17. Разработайте интеграционный тест, проверяющий корректность передачи данных между модулем обучения и модулем валидации модели.
18. Проведите регрессионное тестирование набора ключевых функциональностей после внесения изменений в алгоритм.
19. Создайте чек-лист для проверки качества мобильного ИИ-приложения перед релизом.
20. Визуализируйте матрицу ошибок для многоклассовой классификации и проинтерпретируйте результаты.
21. Оцените качество модели линейной регрессии, рассчитав метрики MAE и MSE на тестовой выборке.
22. Проведите A/B тестирование двух версий модели для определения наиболее эффективной.
23. Протестируйте модель кластеризации на синтетическом наборе данных, рассчитав индекс силуэта.
24. Настройте автоматический запуск юнит-тестов при каждом коммите в репозиторий с помощью GitHub Actions.
25. Сгенерируйте синтетический тестовый датасет для проверки устойчивости модели к шуму во входных данных.
26. Разработайте тесты для мобильного приложения, использующего камеру для захвата изображений и их классификации.
27. Протестируйте работу ИИ-приложения в условиях нестабильного сетевого соединения.
28. Создайте тестовый стенд, имитирующий производственное окружение для проведения E2E-тестов.
29. Проведите стресс-тестирование базы данных, используемой для хранения логов предсказаний модели.
30. Просканируйте веб-приложение с ИИ-компонентом на наличие уязвимостей (например, SQL-инъекций) с помощью специализированных инструментов.
31. Настройте в Postman коллекцию запросов для тестирования API предсказаний модели и запустите ее.
32. Проведите профилирование производительности ИИ-приложения для выявления "узких" мест (bottlenecks) при инференсе.
33. Реализуйте мониторинг задержки (latency) и потребления памяти моделью в реальном времени.

34. Проведите юзабилити-тестирование интерфейса настройки гиперпараметров модели с привлечением 3-5 тестировщиков.
35. Протестируйте совместимость веб-интерфейса для визуализации предсказаний с различными браузерами (Chrome, Firefox, Safari).
36. Напишите и выполните тесты для REST API, проверяющие корректность обработки некорректных входных данных (валидация).
37. Настройте систему оповещения (alerting) при падении точности модели (концепция дрейфа данных) ниже заданного порога.
38. Реализуйте дашборд для визуализации ключевых метрик производительности и качества модели в реальном времени.
39. Проведите тестирование процедуры бекапа и восстановления данных, используемых для переобучения модели.
40. Протестируйте систему логирования всех этапов работы ИИ-пайплайна на корректность и полноту записываемой информации.

Типовые задания для оценки освоения МДК 01.03. Тестирование программных модулей

Теоретические вопросы

1. Определение качества программного обеспечения согласно стандарту ГОСТ Р ИСО/МЭК 25051.
2. Основные метрики качества программного продукта и их назначение.
3. Ключевые отличия модульного, интеграционного, системного и приемочного уровней тестирования.
4. Сущность тестирования, основанного на спецификациях (specification-based testing).
5. Цели и методы тестирования на основе сценариев использования (use case testing).
6. Роль диаграммы причинно-следственных связей в процессе тест-дизайна.
7. Основные виды тестирования производительности и их цели.
8. Процесс регрессионного тестирования и его место в жизненном цикле разработки.
9. Основные этапы жизненного цикла дефекта (bug life cycle).
10. Критерии определения уровня серьезности (severity) дефекта.
11. Компоненты стратегии тестирования программного обеспечения.
12. Структура и содержание тестового плана (test plan).
13. Отличие тестового сценария (test case) от чек-листа (checklist).
14. Назначение тестового пакета (test suite) и задания на тестирование (test run).
15. Понятие шаблонов тестов (test patterns) и их применение.
16. Сущность техники тест-дизайна «Эквивалентное Разбиение» (Equivalence Partitioning).
17. Принцип анализа граничных значений (Boundary Value Analysis).
18. Методика создания и использования таблиц решений (Decision Tables) для тестирования.
19. Процесс тестирования переходов состояний (State Transition Testing).
20. Основные метрики для оценки качества моделей машинного обучения.
21. Компоненты матрицы ошибок (Confusion Matrix) и их интерпретация.
22. Формулы расчета и смысл метрик Precision (Точность) и Recall (Полнота).
23. Недостатки метрики Accuracy (Доля правильных ответов) в условиях несбалансированных данных.
24. Назначение и расчет F1-score (F-мера) как агрегированной метрики.
25. Принцип построения и интерпретации ROC-кривой (Receiver Operating Characteristic curve).
26. Смысл площади под ROC-кривой (AUC-ROC) и ее интерпретация.

27. Ограничения и области применения ROC-кривой и метрики AUC.
28. Основные метрики для оценки моделей регрессии.
29. Формула и интерпретация MAE (Mean Absolute Error).
30. Формула, интерпретация и ограничения MAPE (Mean Absolute Percentage Error).
31. Понятие переобучения (overfitting) и критерии определения оптимальной сложности модели.
32. Основные метрики для оценки качества алгоритмов кластеризации.
33. Инструменты для автоматизации выполнения тестовых сценариев.
34. Принципы интеграции автоматизированных тестов в конвейер CI/CD.
35. Назначение и типы генераторов тестовых данных (data generators).
36. Специфические проблемы и подходы к тестированию мобильных ИИ-приложений.
37. Понятие тестового окружения (test environment) и его ключевые компоненты.
38. Назначение и виды заглушек (test doubles) в тестировании.
39. Цели и отличия End-to-End (E2E) тестирования от других уровней.
40. Методы и цели тестирования отказоустойчивости (failover testing).
41. Подходы к проведению стресс-тестирования (stress testing) систем.
42. Ключевые аспекты тестирования безопасности (security testing) ИИ-приложений.
43. Инструменты для автоматизации тестирования веб-API.
44. Методы мониторинга производительности (performance monitoring) ИИ-приложений.
45. Процесс профилирования (profiling) ИИ-моделей для выявления узких мест.
46. Цели и этапы проведения юзабилити-тестирования (usability testing).
47. Подходы к тестированию совместимости с различными браузерами (cross-browser testing).
48. Основные принципы тестирования API (API testing).
49. Методы мониторинга ИИ-моделей в производственной среде.
50. Системы для визуализации метрик производительности и данных мониторинга.
51. Понятие «дрейфа данных» (data drift) и его влияние на качество предсказаний модели.
52. Метрики, используемые для оценки бизнес-ценности ИИ-модели.
53. Роль тестирования в обеспечении этичности и отсутствия предвзятости (fairness) в ИИ-системах.
54. Принципы тестирования генеративных моделей (generative models).
55. Подходы к тестированию устойчивости (robustness) модели к зашумленным данным.
56. Методика тестирования скорости инференса (времени отклика) модели.
57. Процесс нагрузочного тестирования (load testing) сервиса, использующего ИИ.
58. Принципы проведения A/B тестирования различных версий моделей.
59. Роль системы контроля версий для данных (Data Version Control, DVC) в процессе тестирования.
60. Профессиональные и этические нормы, связанные с тестированием и валидацией ИИ-систем.

Практические задания

1. Разработайте чек-лист для проверки качества программного обеспечения согласно стандарту ГОСТ Р ИСО/МЭК 25051.
2. Определите цели тестирования для модульного, интеграционного и системного уровней для заданного функционального требования.
3. Спроектируйте тестовые случаи для модульного тестирования с использованием техники эквивалентного разбиения.
4. Примените анализ граничных значений для создания тестовых данных поля ввода, принимающего значения от 18 до 65.

5. Разработайте таблицу решений для тестирования логики предоставления скидки на основе условий: статус клиента и сумма покупки.
6. Постройте диаграмму причинно-следственных связей для функции аутентификации пользователя.
7. Создайте тестовый сценарий для проверки работы функции на основе сценария использования «Восстановление пароля».
8. Подготовьте тестовый пакет для проведения регрессионного тестирования после исправления критического дефекта.
9. Проведите оценку метрик качества программного продукта на основе предоставленного отчета о тестировании.
10. Составьте тестовое задание для проверки интеграции между модулем обработки заказов и модулем расчета доставки.
11. Разработайте шаблон тестового сценария для функционального тестирования веб-форм.
12. Спроектируйте тестовые случаи для проверки перехода системы между состояниями «Неактивна», «Активна», «Заблокирована».
13. Рассчитайте метрики Precision, Recall и F1-score на основе предоставленной матрицы ошибок 3x3.
14. Постройте ROC-кривую по заданным значениям TPR и FPR и рассчитайте площадь под кривой (AUC).
15. Проведите анализ влияния изменения порога классификации на метрики Precision и Recall.
16. Реализуйте функцию для вычисления MAE и MAPE на тестовой выборке модели регрессии.
17. Проведите оценку качества кластеризации на синтетическом наборе данных, рассчитав индекс силуэта.
18. Определите оптимальную сложность модели, проанализировав графики ошибок на тренировочной и валидационной выборках.
19. Настройте автоматический запуск юнит-тестов в GitLab CI при каждом пуше в ветку develop.
20. Интегрируйте выполнение тестов производительности в пайплайн Jenkins.
21. Создайте генератор синтетических тестовых данных для проверки устойчивости модели к аномальным значениям.
22. Разработайте тестовый сценарий для проверки работы мобильного ИИ-приложения в офлайн-режиме.
23. Подготовьте тестовое окружение с использованием Docker-контейнеров для изоляции зависимостей.
24. Создайте заглушки для внешних сервисов в интеграционных тестах модели, зависящей от данных из API.
25. Проведите End-to-End тестирование пайплайна машинного обучения от загрузки сырых данных до получения предсказания.
26. Реализуйте сценарий тестирования отказоустойчивости при недоступности базы данных с фичами.
27. Проведите стресс-тестирование веб-сервиса классификации изображений, постепенно увеличивая количество параллельных запросов.
28. Просканируйте веб-приложение с помощью специализированного инструмента для выявления уязвимостей XSS.
29. Создайте коллекцию запросов в Postman для тестирования REST API сервиса предсказаний.
30. Настройте мониторинг времени отклика (latency) модели в продакшн-окружении с помощью Prometheus.

31. Проведите профилирование памяти и CPU ИИ-приложения с помощью cProfile или аналога.
32. Разработайте сценарий юзабилити-тестирования интерфейса настройки гиперпараметров модели.
33. Проведите тестирование совместимости веб-интерфейса дашборда с моделями в браузерах Chrome, Firefox и Safari.
34. Напишите тесты для REST API, проверяющие корректность HTTP-статусов и структуры JSON-ответов при ошибках.
35. Настройте дашборд в Grafana для визуализации ключевых метрик качества модели (Accuracy, F1-score) в реальном времени.
36. Реализуйте систему алертинга при значительном изменении распределения входных данных (дрейф данных).
37. Проведите тестирование процедуры бекапа и восстановления обучающих данных.
38. Протестируйте корректность работы системы логирования на всех этапах ML-пайплайна.
39. Проведите аудит тестового покрытия кода модуля предобработки данных и составьте отчет.
40. Разработайте план регрессионного тестирования для ИИ-системы после обновления версии фреймворка машинного обучения.