

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Узунов Федор Владимирович

Должность: Ректор

Дата подписания: 24.09.2026 14:09:48

Уникальный программный ключ:

fd935d10451b860e912264c0378f8448453bfdb607f94788008e20877a6bcbf5

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
«ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ»
«УНИВЕРСИТЕТ ЭКОНОМИКИ И УПРАВЛЕНИЯ»**

Факультет экономики, управления и юриспруденции

Кафедра управления и бизнес-информатики



УТВЕРЖДАЮ

Проректор по учебно-методической работе

/ Г.П. Узунова

«02» февраля 2026 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ИНФОРМАЦИОННЫХ
СИСТЕМ**

Направление подготовки

09.03.02 «Информационные системы и технологии»

Профиль

«Специалист информационных систем и технологий»

Квалификация выпускника: бакалавр

Для всех
форм обучения

Симферополь, 2026 г.

1. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ

Средства оценивания в ходе текущего контроля:

- устные опросы в ходе семинарских занятий;
- рефераты;
- тестирование;
- практические задания, выполняемые в ходе семинарского (практического) занятия или рекомендуемые для самостоятельной работы.

Коды компетенции	Результаты освоения ОПОП	Перечень планируемых результатов обучения по дисциплине
ОПК-8	ОПК-8. Способен применять математические модели, методы и средства проектирования информационных и автоматизированных систем	ОПК-8.1. Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения. ОПК-8.2. Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули. ОПК-8.3. Владеть: языком программирования; навыками отладки и тестирования работоспособности программы.

1.1 Вопросы к текущему контролю

1. Дайте определение инструментальным средствам информационных систем (ИС). Какова их основная цель?
2. На какие основные группы классифицируются инструментальные средства ИС? Приведите примеры для каждой группы.
3. Что такое CASE-средства? Назовите ключевые функции современных CASE-технологий.
4. В чем заключается принципиальное различие между средствами разработки прикладного программного обеспечения и средствами проектирования архитектуры ИС?
5. Опишите жизненный цикл информационной системы. На каких этапах ЖЦ применяются различные виды инструментальных средств?
6. Что такое интегрированная среда разработки (IDE)? Перечислите обязательные компоненты типичной IDE.
7. Реляционные СУБД: понятие, базовые принципы работы, язык SQL как основной инструмент взаимодействия.
8. NoSQL-СУБД: причины возникновения, основные типы (ключ-значение, документные, графовые, колоночные) и сферы применения.
9. Инструменты администрирования БД: средства мониторинга производительности, резервного копирования и восстановления данных.
10. Что такое ORM (Object-Relational Mapping)? Назовите популярные библиотеки/фреймворки ORM для одного из языков программирования (например, Hibernate для Java или Entity Framework для C#).
11. Унифицированный язык моделирования UML: назначение и структура. Какие типы диаграмм используются на этапе анализа требований к ИС?
12. Методологии функционального моделирования IDEF0 и DFD: в чем разница подходов и для решения каких задач они предназначены?
13. Инструменты визуального прототипирования интерфейсов (Figma, Sketch, Axure): роль в процессе разработки пользовательского опыта (UX).

14. Понятие метамоделирования. Для чего создаются предметно-ориентированные языки моделирования (DSL)?
15. Фреймворк: определение, отличие от библиотеки. Преимущества использования каркасного подхода при создании корпоративных ИС.
16. Системы контроля версий (VCS): Git как стандарт де-факто. Объясните разницу между локальными, централизованными и распределенными VCS.
17. Платформы быстрой разработки приложений (*Low-code / No-code*): преимущества, недостатки и ограничения применимости в корпоративном сегменте.
18. Специализированные среды разработки: приведите примеры инструментов для мобильной разработки (Android Studio, Xcode) и веб-разработки.
19. Инструментальные средства модульного тестирования (*Unit Testing*): JUnit, PyTest, NUnit. Принцип их интеграции в процесс сборки проекта.
20. Инструменты автоматизации функционального тестирования UI (Selenium WebDriver, Cypress): архитектура и принцип работы.
21. Профилировщики кода и анализаторы покрытия тестами: зачем они нужны и как помогают оптимизировать производительность ИС?
22. Статические анализаторы безопасности кода (SAST) и инструменты поиска уязвимостей (DAST, SCA). Примеры таких решений.
23. Системы автоматической сборки проектов (*Build Systems*): Maven, Gradle, MSBuild. Понятие артефакта сборки.
24. Системы непрерывной интеграции (CI) и доставки (CD): Jenkins, GitLab CI/CD, GitHub Actions. Основные этапы классического пайплайна.
25. Контейнеризация: Docker как стандартный инструмент упаковки приложений со всеми зависимостями.
26. Системы оркестрации контейнеров (Kubernetes): задачи масштабирования, балансировки нагрузки и самовосстановления сервисов.
27. Инструменты для описания RESTful API (Swagger/OpenAPI, Postman): их роль в контрактном подходе к разработке микросервисов.
28. Сервисные шины предприятия (ESB) и брокеры сообщений (RabbitMQ, Apache Kafka, ActiveMQ) как инструментальные средства интеграции гетерогенных систем.
29. Средства имитационного моделирования бизнес-процессов (BPMN-редакторы, например, Camunda Modeler): связь исполняемых движков с нотацией процессов.
30. Представьте архитектуру современного стека инструментальных средств для сквозной разработки облачной информационной системы (от этапа сбора требований до развертывания в эксплуатацию). Укажите по одному ключевому инструменту для каждого этапа жизненного цикла и обоснуйте выбор связки этих инструментов.

1.2 Темы рефератов:

1. Эволюция CASE-средств: от простых графических редакторов до комплексных систем управления архитектурой предприятия.
2. Сравнительный анализ современных IDE (Visual Studio, IntelliJ IDEA, VS Code): производительность, экосистема расширений и поддержка языков программирования.
3. Git как стандарт де-факто: архитектура распределенной системы контроля версий и паттерны командной работы (*GitFlow, Trunk-Based Development*).
4. Платформы Low-code/No-code в корпоративном сегменте: возможности кастомизации, риски безопасности и сценарии применения для быстрой разработки ИС.
5. Инструменты объектно-реляционного отображения (ORM): сравнительный анализ Hibernate (Java) и Entity Framework Core (C#), проблема «N+1 запросов».
6. Роль Docker в современной разработке: изоляция сред, управление зависимостями и концепция Infrastructure as Code на уровне контейнеров.
7. Оркестрация микросервисов с помощью Kubernetes: автоматизация развертывания, масштабирования и управления контейнеризированными приложениями.

8. Средства моделирования бизнес-процессов (BPMN 2.0): от нотации к исполняемым движкам (на примере Camunda или Bizagi).
9. Инструментарий для проектирования RESTful API: спецификации OpenAPI (Swagger) и инструменты автоматизированного тестирования контрактов (Postman, Dredd).
10. Брокеры сообщений в высоконагруженных системах: сравнительный анализ архитектур RabbitMQ, Apache Kafka и Redis Streams.
11. Статический анализ кода (SAST) и поиск уязвимостей компонентов (SCA): интеграция SonarQube, Checkmarx или Snyk в цикл непрерывной интеграции.
12. Автоматизация функционального тестирования UI: эволюция инструментов от Selenium WebDriver к Playwright и Cypress.
13. Современные СУБД класса NewSQL: попытка объединить масштабируемость NoSQL и транзакционную надежность SQL (на примере CockroachDB или YugabyteDB).
14. Инструментальные средства поддержки методологии DevSecOps: встраивание проверок безопасности на всех этапах жизненного цикла ПО.
15. Профилировщики производительности (Profilers): методы поиска узких мест (bottlenecks) в коде и управлении памятью приложений.
16. Системы мониторинга и трейсинга распределенных систем: стек Prometheus + Grafana против коммерческих решений (Datadog, New Relic).
17. Инструменты управления ИТ-инфраструктурой (IaC): сравнение декларативного подхода Terraform и императивного подхода Ansible.
18. Специализированные инструментальные средства мобильной разработки: особенности экосистем Android Studio и Xcode, симуляторы и профайлеры устройств.
19. Искусственный интеллект как инструмент разработчика: применение больших языковых моделей (LLM) для генерации кода, автодополнения (GitHub Copilot) и рефакторинга.
20. Отечественные инструментальные средства разработки: обзор российских аналогов зарубежных платформ (СУБД, среды разработки, решения для виртуализации) в рамках обеспечения технологического суверенитета.

1.3 Тестовые задания

1. Какой из следующих инструментов используется для 3D-моделирования?

- а) Microsoft Word
- б) AutoCAD (Правильный ответ: б)
- в) Adobe Photoshop
- г) Notepad

2. Какой этап является первым в процессе проектирования?

- а) Исследование (Правильный ответ: а)
- б) Моделирование
- в) Тестирование
- г) Документация

3. Какой из следующих типов чертежей используется для представления архитектурных объектов?

- а) Схема
- б) План (Правильный ответ: б)
- в) Разрез
- г) Модель

4. Какое программное обеспечение предназначено для автоматизации производства?

- а) CAD
- б) CAM (Правильный ответ: б)
- в) CAE

г) PDM

5. Какой из следующих процессов не относится к автоматизированному проектированию?

а) Чертежи

б) Ручное рисование (Правильный ответ: б)

в) Моделирование

г) Анализ

6. Какой из следующих этапов включает в себя проверку на соответствие требованиям?

а) проектирование

б) верификация (Правильный ответ: б)

в) документация

г) моделирование

7. Какой этап жизненного цикла изделия следует непосредственно после создания рабочей конструкторской документации?

а) утилизация

б) производство (Правильный ответ: б)

в) маркетинговое исследование

г) эскизное исследование

8. Что обозначает аббревиатура CAD?

а) Computer-Aided Design

б) Computer-Aided Drafting

в) Оба варианта верны (Правильный ответ: в)

г) Ничего не подходит

9. Какой формат чаще всего используется для обмена трехмерными моделями между разными CAD-системами?

а) .step (Правильный ответ: а)

б) .exe

в) .docx

г) .mp3

10. Какой из следующих видов анализа не выполняется в CAE-системах?

а) Прочностной расчет

б) Расчет гидродинамики

в) Создание управляющей программы для ЧПУ (Правильный ответ: в)

г) Тепловой анализ

11. Какие из следующих программ относятся к CAD?

а) AutoCAD

б) SolidWorks

в) Excel

г) CATIA

Ответ: а, б, г.

12. Какие преимущества дает автоматизированное проектирование?

а) Ускорение процесса

б) Увеличение затрат

в) Повышение точности

г) Снижение ошибок

Ответ: а, в, г.

13. Какие из следующих этапов входят в процесс разработки проекта?

а) Анализ требований

б) Создание прототипа

в) Подбор оборудования

г) Тестирование

Ответ: а, б, г.

14. Какие системы проектирования используются для управления данными в проектировании?

- а) PDM
- б) CAD
- в) ERP
- г) CAM

Ответ: а, в.

15. Какие из следующих методов используются в процессе верификации?

- а) Моделирование
- б) Тестирование
- в) Опрос пользователей
- г) Сравнение с эталоном

Ответ: а, б, г.

16. Какие из следующих типов чертежей могут быть использованы в проектировании?

- а) Разрез
- б) Схема
- в) График
- г) План

Ответ: а, б, г.

17. Какие из следующих технологий могут быть использованы в автоматизированном проектировании?

- а) 3D-печать.
- б) Виртуальная реальность.
- в) Искусственный интеллект.
- г) Ручное рисование.

Ответ: а, б, в.

18. Какие из следующих форматов файлов предназначены для обмена трехмерными геометрическими данными между разными CAD-системами?

- а) STEP.
- б) IGES.
- в) DWG.
- г) STL.

Ответ: а, б.

19. Какие из перечисленных задач относятся к компетенции CAE-систем?

- а) Прочностной расчет методом конечных элементов.
- б) Создание управляющей программы для станка с ЧПУ.
- в) Расчет гидрогазодинамики.
- г) Тепловой анализ.

Ответ: а, в, г.

20. Какие из следующих видов информации содержит рабочая конструкторская документация?

- а) Сборный чертеж изделия.
- б) Спецификация компонентов.
- в) Бюджет маркетинговой кампании.
- г) Технические условия на изготовление.

Ответ: а, б, г.

21. Установить соответствие между термином и определением:

1. AutoCAD	А) 3D-моделирование
------------	---------------------

2. SolidWorks	Б) Чертежи и схемы
3. CATIA	В) Системы для проектирования в аэрокосмической отрасли
4. Revit	Г) Архитектурное проектирование

22. Установить соответствие между термином и определением:

1. Исследование	А) Сбор данных
2. Проектирование	Б) Создание модели
3. Тестирование	В) Проверка работоспособности
4. Документация	Г) Оформление результатов

23. Установить соответствие между термином и определением:

1. Схема	А) Показать взаимосвязи
2. Чертеж	Б) Подробное изображение
3. План	В) Вид сверху
4. Разрез	Г) Внутреннее строение

24. Установить соответствие между термином и определением:

1. CAD	А) Компьютерное проектирование
2. CAM	Б) Автоматизация производства
3. CAE	В) Анализ и верификация
4. PDM	Г) Управление данными

25. Установить соответствие между термином и процессом:

1. Блок-схема	А) Графическое представление алгоритма
2. Моделирование	Б) Создание абстрактной модели
3. Прототип	В) Первоначальная версия продукта
4. Верификация	Г) Проверка соответствия требованиям

26. Установить соответствие между термином и определением:

1. Семафор	А) Участок кода с ограниченным доступом
2. Мьютекс	Б) Примитив синхронизации с счетчиком
3. Критическая секция	В) Ситуация взаимного ожидания процессов
4. Взаимная блокировка	Г) Механизм взаимоисключения

27. Установить соответствие между термином и определением:

1. Драйвер устройства	А) Временное хранение данных
2. Буферизация	Б) Программа управления устройством
3. Спулинг	В) Механизм уведомления процессора
4. Прерывание	Г) Организация очереди ввода-вывода

28. Установить соответствие между термином и определением:

1. Аутентификация	А) Проверка прав доступа
2. Авторизация	Б) Проверка личности
3. Шифрование	В) Преобразование данных в защищенный вид
4. Аудит	Г) Регистрация действий

29. Установить соответствие между термином и определением:

1. Однопользовательская	А) Работает с несколькими пользователями
2. Многопользовательская	Б) Обработывает задачи с жесткими временными ограничениями
3. Реального времени	В) Использует несколько компьютеров
4. Распределенная	Г) Рассчитана на одного пользователя

30. Установить соответствие между термином и определением:

1. TCP	А) Преобразование доменных имен
2. IP	Б) Надежная передача данных
3. DNS	В) Назначение IP-адресов
4. DHCP	Г) Адресация пакетов

1.4 Практические задания

Практическая работа 1. Реализация метрических методов классификации: метод ближайшего соседа и метод парzenов.

В качестве реальной задачи взять задачу классификации происхождения вина (3 класса) по трем признакам: общее содержание фенолов, содержание флавоноидных и нефлавоноидных фенолов. Данные хорошо разделимы по этим трем признакам.

Практическая работа 2. Реализация метрических методов классификации: отбор эталонных объектов (STO LP).

Реализовать отбор эталонных объектов на основе STOLP.

Отбор эталонных объектов STOLP Вход: X_ℓ — обучающая выборка; δ — порог фильтрации выбросов; ℓ_0 — допустимая доля ошибок; Выход: Множество опорных объектов $\Omega \subseteq X_\ell$; 1: для всех $x_i \in X_\ell$ проверить, является ли x_i выбросом: 2: если $M(x_i, X_\ell) < \delta$ то 3: $X_{\ell-1} := X_\ell \setminus \{x_i\}$; $\ell := \ell - 1$; 4: Инициализация: взять по одному эталону от каждого класса: $\Omega := \emptyset$; $\arg \max_{x_i \in X_\ell} y M(x_i, X_\ell) \neq y \in Y$; 5: пока $\Omega \neq X_\ell$; 6: Выделить множество объектов, на которых алгоритм $a(u; \Omega)$ ошибается: $E := \{x_i \in X_\ell \setminus \Omega : M(x_i, \Omega) < 0\}$; 7: если $|E| < \ell_0$ то 8: выход; 9: Присоединить к Ω объект с наименьшим отступом: $x_i := \arg \min_{x \in E} M(x, \Omega)$; $\Omega := \Omega \cup \{x_i\}$;

Практическая работа 3-4. Реализация метода опорных векторов.

Реализовать метод опорных векторов для случая линейно разделимой выборки. Пусть выборка линейно разделима, то есть существует некоторая гиперплоскость, разделяющая классы -1 и $+1$. Тогда в качестве алгоритма классификации можно использовать линейный пороговый классификатор:

$$a(x) = \text{sign}(\langle \vec{w}, x \rangle - b) = \text{sign}(\sum_{i=1}^{\ell} w_i x_i - b) \quad a(\vec{x}) = \text{sign}(\langle \vec{w}, \vec{x} \rangle - b) = \text{sign}\left(\sum_{i=1}^{\ell} w_i x_i - b\right)$$

где $x = (x_1, \dots, x_n)$, $\vec{x} = (x_1, \dots, x_n)$ — вектор значений признаков объекта, а $\vec{w} = (w_1, \dots, w_n) \in \mathbb{R}^n$, $\vec{w} = (w_1, \dots, w_n) \in \mathbb{R}^n$ и $b \in \mathbb{R}$ — параметры гиперплоскости.

Но для двух линейно разделимых классов возможны различные варианты построения разделяющих гиперплоскостей. Метод опорных векторов выбирает ту гиперплоскость, которая максимизирует отступ между классами.

Практическая работа 5-6. Реализация линейных и нелинейных моделей регрессии.

Реализовать модель линейной регрессии на Python. Решить задачу классификации на основе логистической регрессии и оценить точность классификации.

Практическая работа 7. Реализация нейрона и полносвязной нейронной сети.

Реализовать нейрон и полносвязную нейронную сеть на Python.

Нейронные сети — это математические модели, которые по своей структуре и принципу работы во многом напоминают сети нервных клеток в мозге. Они состоят из соединенных между собой нейронов — простых вычислительных элементов. Нейрон получает на вход определенные данные, выполняет над ними вычисления согласно своей функции активации, и передает результат на выход, который подается на входы следующих нейронов. Для реализации следует использовать TensorFlow в Python, создать и обучить простую нейронную сеть.

Практическая работа 8. Реализация деревьев решений.

Построить дерево решений на основе датасета. Обучить решающее дерево по датасету с данными о заемщиках банка (задача кредитного скоринга).

Практическая работа 9. Реализация случайного леса.

Реализовать случайный лес на Python.

Загрузите ваши данные, в заданном наборе данных определите случайную выборку, далее алгоритм построит по выборке дерево решений. Дерево строится, пока в каждом листе не более n объектов, или пока не будет достигнута определенная высота. Затем будет получен результат прогнозирования из каждого дерева решений. На этом этапе голосование будет проводиться для каждого прогнозируемого результата: мы выбираем лучший признак, делаем разбиение в дереве по нему и повторяем этот пункт до исчерпания выборки. В конце результата прогноза с наибольшим количеством голосов. Это и есть окончательный результат прогнозирования. Для реализации алгоритма Random Forest необходима формула итогового классификатора $a(x)$:

$$a(x) = \frac{1}{N} \sum_{i=1}^N b_i(x)$$

Где N – количество деревьев, i – счетчик для деревьев, b – решающее дерево, x – сгенерированная нами на основе данных выборка.

Практическая работа 10. Реализация линейных ансамблей.

Реализовать бустинг на Python. Градиентный бустинг добавляет базовые модели в ансамбль последовательно, однако вместо обучения моделей с учетом весов на основе ошибок предшественников, в данном случае модели обучаются на остаточных ошибках (residual errors), допущенных предыдущими моделями.

Формулы для расчетов регрессии:

Функция потерь:

$$L(y_i, F(x_i)) = \frac{1}{2} (y_i - F(x_i))^2$$

Остатки:

$$r_{ik} = - \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x)=F_{m-1}(x)} = y_i - F_{m-1}(x_i)$$

Практическая работа 11. Реализация байесовской классификации.

Реализовать байесовскую классификацию в Python.

Формулы для расчетов:

Прогноз случайного леса для регрессии:

$$\hat{f}_{rf}^B(x) = \frac{1}{B} \sum_{b=1}^B T_b(x)$$

Прогноз случайного леса для классификации:

$$\hat{C}_{rf}^B(x) = \text{majority vote}[\hat{C}_b(x)]_1^B$$

Практическая работа 12-13. Реализация бустинга и бэггинга.

Реализовать линейные ансамбли на Python. Использовать такие методы как стекинг, бэггинг и бустинг. Реализовать байесовскую классификацию на Python.

Практическая работа 14-16. Реализация методов кластеризации: kMeans и иерархическая кластеризация, DBScan, кейс.

Реализовать алгоритм kMeans и алгоритм иерархической кластеризации на Python.

Реализовать алгоритм DBScan на Python. Сравнить алгоритмы кластеризации на примере реализации кейса «Сегментация клиентов».

2. КРИТЕРИИ ОЦЕНИВАНИЯ ПРИ ПРОВЕДЕНИИ ТЕКУЩЕГО КОНТРОЛЯ

Вид контроля	Наименование работы	Наименование оценочных средств	Шкала оценивания
Текущий контроль	- Вопросы для обсуждения на занятиях; - Устные опросы по ранее изученному материалу;	Оценка выступлений на практическом (семинарском)	отлично хорошо

	- Письменные работы: рефераты, тестовые задания; - Практические задания; - Рефераты и доклады по темам (вопросам), вынесенным на самостоятельную работу.	занятия, проверка заданий и аудиторных работ, устный опрос, оценивание докладов, рефератов	удовлетворительно неудовлетворительно
--	--	--	--

Критерии оценивания устных ответов обучающихся

Шкала оценивания	Характеристика оценивания
отлично	оценивается ответ, который показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа.
хорошо	оценивается ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.
удовлетворительно	оценивается ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа.
неудовлетворительно	оценивается ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа.

Критерии оценивания работы обучающихся на практических и семинарских занятиях

Шкала оценивания	Показатели	Критерии
Отлично	1. Полнота выполнения практического и тестового задания (полнота ответа); 2. Своевременность	Задание решено самостоятельно. При этом составлен правильный алгоритм решения задания, в логических рассуждениях, в выборе формул и решении нет ошибок, получен верный ответ, задание решено рациональным способом. Дан правильный и исчерпывающий ответ на

Шкала оценивания	Показатели	Критерии
	выполнения задания; 3. Последовательность и рациональность выполнения практического задания	поставленные теоретические и тестовые вопросы, в которых обучающийся показал всестороннее системное знание программного материала, усвоение основной и дополнительной литературы, четкое владение понятийным аппаратом.
Хорошо	(логичность и четкость ответа); 4. Правильность ответов на вопросы; 5. Самостоятельность решения (владение дополнительным материалом); 6. Знание нормативно-законодательной базы и терминологии курса	Задание решено с помощью преподавателя. При этом составлен правильный алгоритм решения задания, в логическом рассуждении и решении нет существенных ошибок; правильно сделан выбор формул для решения; есть объяснение решения, но задание решено нерациональным способом или допущено не более двух несущественных ошибок, получен верный ответ. На поставленные теоретические и тестовые вопросы, при которых обучающийся показал достаточный уровень знаний основного программного материала: освоение информации лекционного курса и учебных пособий, овладение понятийным аппаратом, методикой исследований при попытке анализа различных ситуаций.
Удовлетворительно		Задание решено с подсказками преподавателя. Задание решено в общем виде. Обучающийся показал средний уровень знаний основного программного материала, но не мог убедительно аргументировать свой ответ, ошибся в использовании понятийного аппарата, показал недостаточные знания литературных источников.
Неудовлетворительно		Задание не решено. Обучающийся продемонстрировал значительные пробелы в знаниях основного программного материала, не аргументировал свой ответ, показал неудовлетворительные знания понятийного аппарата и специальной литературы.

Критерии оценивания рефератов

Средство контроля	Критерии оценивания	Шкала оценивания
Реферат	Выполнены все требования к написанию и защите реферата: обозначена проблема и обоснована её актуальность, сделан краткий анализ различных точек зрения на рассматриваемую проблему и логично изложена собственная позиция, сформулированы выводы, тема раскрыта полностью, выдержан объём, соблюдены требования к внешнему оформлению, даны правильные ответы на дополнительные вопросы. Реферат раскрывает поднятую проблематику в полном объеме.	отлично
	Основные требования к реферату и его защите выполнены, но при этом допущены недочёты. В частности, имеются неточности в изложении материала; отсутствует логическая последовательность в суждениях; не выдержан объём реферата;	хорошо

	имеются упущения в оформлении; на дополнительные вопросы при защите даны неполные ответы. В реферате имеются неточности и предметная область выступления раскрыта не в полной мере.	
	Имеются существенные отступления от требований к реферированию. В частности: тема освещена лишь частично; допущены фактические ошибки в содержании реферата или при ответе на дополнительные вопросы; во время защиты отсутствует вывод. В реферате не в полной степени раскрыт понятийный аппарат, имеются существенные неточности в процессе формирования выводов.	удовлетворительно
	Тема реферата не раскрыта, обнаруживается существенное непонимание проблемы. Тема реферата не раскрыта или выполнена не по существу ранее поставленного вопроса. Реферат не сдан / доклад не сделан.	неудовлетворительно

Критерии оценивания тестов

Средство контроля	Критерии оценивания – процент положительных ответов	Шкала оценивания
Тестирование	90-100	отлично
	70-89	хорошо
	40-69	удовлетворительно
	< 39	неудовлетворительно

3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Средства оценивания в ходе промежуточной аттестации:

- вопросы для экзамена;
- тестовые задания к экзамену.

Коды компетенции	Результаты освоения ОПОП	Перечень планируемых результатов обучения по дисциплине
ОПК-8	ОПК-8. Способен применять математические модели, методы и средства проектирования информационных и автоматизированных систем	ОПК-8.1. Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения. ОПК-8.2. Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули. ОПК-8.3. Владеть: языком программирования; навыками отладки и тестирования работоспособности программы.

3.1. Вопросы к экзамену:

1. Прикладные задачи машинного обучения: примеры и особенности.
2. Линейный классификатор: математическая модель, алгоритм обучения.

3. Стохастический градиентный спуск: принципы работы, настройка параметров.
4. Нейронные сети: архитектура, функции активации, обучение.
5. Метрические методы классификации: метод ближайшего соседа, парзеновское окно.
6. Метрические методы регрессии: выбор метрики, особенности применения.
7. Метод опорных векторов: геометрическая интерпретация, ядра, гиперпараметры.
8. Линейные модели регрессии: оценка коэффициентов, предпосылки метода.
9. Нелинейная регрессия: полиномиальная регрессия, сплайны, GAM.
10. Качество классификации: метрики (accuracy, precision, recall, F1-score).
11. Матрица ошибок (confusion matrix) и её анализ.
12. ROC-кривая и AUC-ROC: построение и интерпретация.
13. Отбор признаков: фильтрация, обёртка, встроенные методы.
14. Логические методы классификации: деревья решений, правила.
15. Случайный лес: принцип работы, настройка гиперпараметров.
16. Бустинг: AdaBoost, Gradient Boosting, XGBoost — сравнение и особенности.
17. Бэггинг: Random Forest, Pasting — принципы и отличия.
18. Стекинг: реализация и сравнение с другими ансамблями.
19. Байесовская классификация: теорема Байеса, Naive Bayes.
20. Кластеризация: kMeans — этапы алгоритма, выбор числа кластеров.
21. Иерархическая кластеризация: типы связей, дендрограммы.
22. DBScan: работа с шумом и кластерами произвольной формы.
23. Частичное обучение: определение, алгоритмы (self-training, co-training).
24. Инкрементное обучение: задачи, алгоритмы (Hoeffding Tree), отличия от пакетного.
25. Онлайн-обучение: принципы, алгоритмы (Perceptron Online), области применения.
26. Обучение ранжированию: задачи поиска и рекомендаций, метрики (NDCG).
27. RankNet и LambdaMART: архитектура и особенности обучения ранжированию.
28. Модели внимания: self-attention механизм, архитектура Transformer.
29. Трансформеры в NLP: BERT, GPT — принцип работы и преимущества.
30. Реализация случайного леса на Python: этапы построения модели.
31. Реализация линейных ансамблей на Python.
32. Реализация деревьев решений на Python без библиотек.
33. Реализация нейрона и полносвязной нейронной сети на Python с нуля.
34. Реализация метода ближайшего соседа на Python с нуля.
35. Реализация парзеновского окна на Python с нуля.
36. Реализация метода опорных векторов на Python без библиотек ML.
37. Реализация линейной регрессии методом наименьших квадратов вручную.
38. Реализация байесовского классификатора на синтетических данных.
39. Реализация kMeans вручную без готовых библиотек.
40. Визуализация результатов кластеризации: PCA, t-SNE.
41. Оценка качества кластеризации: silhouette score, Calinski-Harabasz index.
42. Применение ансамблирования для повышения качества моделей.
43. Сравнение метрических методов с параметрическими моделями.
44. Влияние нормализации признаков на результаты метрических методов.
45. Мультиколлинеарность признаков: диагностика и методы борьбы.
46. Регуляризация в линейных моделях: Ridge, Lasso, ElasticNet.
47. Диагностика переобучения нейронных сетей и методы борьбы с ним.
48. Функции потерь в машинном обучении: примеры для классификации и регрессии.
49. Обучение с подкреплением: основные понятия и отличия от supervised learning.
50. Пайплайн машинного обучения: этапы построения и автоматизации.
51. Применение машинного обучения для анализа социальных сетей.
52. Анализ тональности текстов на Python: инструменты и подходы.
53. Кластеризация текстовых данных: TF-IDF + kMeans/DBScan.
54. Обработка и предобработка текстовых данных для ML.

55. Применение машинного обучения для анализа временных рядов.
56. Рекомендательные системы: коллаборативная фильтрация и контентные методы.
57. Современные библиотеки для машинного обучения в Python (scikit-learn, TensorFlow, PyTorch).
58. Визуализация процесса обучения моделей ML (learning curves, feature importance).
59. Автоматизация подбора гиперпараметров моделей (GridSearchCV, RandomizedSearchCV).
60. Современные тренды в машинном обучении (2024–2025): трансформеры, AutoML, объяснимый ИИ (XAI).

3.2. Задания для экзамена:

Задание 1

Реализовать на Python метрический классификатор по методу ближайшего соседа (kNN) для задачи классификации рукописных цифр (MNIST).

Задание 2

Реализовать на Python метод опорных векторов (SVM) с линейным и радиально-базисным ядром для задачи классификации.

Задание 3

Построить и обучить на Python полносвязную нейронную сеть для бинарной классификации (спам/ не спам).

Задание 4

Создать и обучить на Python случайный лес для задачи классификации, вывести важность признаков.

Задание 5

Реализовать на Python дерево решений для задачи регрессии, визуализировать полученное дерево.

Задание 6

Построить на Python ансамбль из базовых моделей с помощью бустинга (AdaBoost или Gradient Boosting).

Задание 7

Реализовать на Python алгоритм kMeans для кластеризации набора данных, подобрать оптимальное число кластеров.

Задание 8

На Python реализовать иерархическую кластеризацию и построить дендрограмму.

Задание 9

Реализовать на Python алгоритм DBScan для поиска кластеров произвольной формы.

Задание 10

Построить на Python байесовский классификатор (Naïve Bayes) для анализа тональности текстов.

Задание 11

Реализовать на Python линейную регрессию методом наименьших квадратов для задачи прогнозирования.

Задание 12

Построить и обучить на Python нейронную сеть для задачи многоклассовой классификации.

Задание 13

Реализовать на Python парзеновское окно для задачи классификации.

Задание 14

Импортируйте набор данных diamonds. Для загрузки используйте библиотеку seaborn и команду `seaborn.load_dataset('diamonds')`. Сколько бриллиантов органики Very Good в этом наборе? За огранку отвечает столбец 'cut'. Выведите ответ.

Задание 15

Импортируйте набор данных `fmri`. Для загрузки используйте библиотеку `seaborn` и команду `seaborn.load_dataset('fmri')`. Ответьте, сколько строк в этом наборе. Выведите ответ.

Задание 16

Импортируйте набор данных `diamonds`. Для загрузки используйте библиотеку `seaborn` и команду `seaborn.load_dataset('diamonds')`. Отсортируйте набор по возрастанию цены. Выберите в полученном наборе строку с индексом 78. Чему равно значение столбца `price` в этой строке?

Задание 17

Импортируйте набор данных `diamonds`. Для загрузки используйте библиотеку `seaborn` и команду `seaborn.load_dataset('diamonds')`. Отберите только бриллианты с огранкой `Ideal`. Сколько стоит третий бриллиант в полученном наборе? За огранку отвечает столбец `'cut'`, за цену столбец `'price'`. Выведите ответ.

Задание 18

Импортируйте набор данных `diamonds`. Для загрузки используйте библиотеку `seaborn` и команду `seaborn.load_dataset('diamonds')`. Сколько бриллиантов огранки `Premium` в этом наборе? За огранку отвечает столбец `'cut'`. Выведите ответ.

Задание 19

Импортируйте набор данных `diamonds`. Для загрузки используйте библиотеку `seaborn` и команду `seaborn.load_dataset('diamonds')`. Отберите только бриллианты цвета «J» и «Premium». Какова стоимость третьего бриллианта в наборе? За цвет отвечает столбец `'color'`, за огранку `'cut'`.

Задание 20

Импортируйте набор данных `brain_networks`. Для загрузки используйте библиотеку `seaborn` и команду `seaborn.load_dataset('brain_networks')`. Ответьте какое значение находится в ячейке датасета с индексами 134,7? Выведите ответ.

Задание 21

Импортируйте набор данных `diamonds`. Для загрузки используйте библиотеку `seaborn` и команду `seaborn.load_dataset('diamonds')`. Отберите только бриллианты, весящие меньше 1-х карата, огранки «Ideal». Какова максимальная стоимость бриллианта из этого набора? За вес отвечает столбец `'carat'`, за огранку `'cut'`, за стоимость `'price'`.

Задание 22

Откройте Jupiter notebook, создайте файл `ekz.ipynb`, подключите библиотеки `pumpru` и `pandas` и прочтите данные из файла `StudentsPerformance.csv` в датафрейм `sp`. Получите распределение числовых величин датафрейма `sp` используя базовые методы визуализации библиотеки `pandas`. Импортируйте библиотеку `Seaborn`. Используя интерфейс для подгонки регрессионных моделей к условным подмножествам набора данных выведите график успеваемости по математике и чтению с разделением по полу.

Задание 23

Загрузите модули библиотек `pumpru` и `matplotlib`, а также модули библиотеки `Scikit-learn`, необходимые для работы с древом решений. Из `sklearn.datasets` загрузите тестовый набор `iris`. Изучите структуру полученного набора данных. В наборе 150 экземпляров ("строк" в привычной нам терминологии). Каждому экземпляру соответствует четыре числовых атрибута (длина чашелистика в см; ширина чашелистика в см; длина лепестка в см; ширина лепестка в см) и класс ('setosa', 'versicolor', 'virginica'). Т.е. данный набор позволяет решить задачу классификации, отнесения растения "ирис" исходя из ширины лепестков, длины чашелистика и т.д. к одному из трех классов. Создайте демонстративный `Decision Tree` классификатор, используя одноимённый класс из библиотеки `sklearn` и сохраните его в переменную `dt`. Для разделения множества на тестовое и тренировочное используйте метод `train_test_split`. Отобразите полученное дерево решений графически, используйте метод `plot_tree`.

Задание 24

Загрузите данные из встроенного набора `load_digits`. Данные представляют собой набор из 1797 рукописных изображений цифр размерами 8 на 8. Каждая картинка представлена массивом из 64 значений цвета для каждой ячейки изображения. Обучите дерево решений и получите предсказание на тестовой выборке. Визуализируйте предсказание, укажите точность для полученного дерева.

Задание 25

Загрузите встроенный набор `boston`. Набор данных `boston` представляет собой словарь с ключами `data`, `target`, `feature_names` и `DESCR`. Ключу `data` соответствует массив списков: `{'data': array([[6.3200e-03, 1.8000e+01, 2.3100e+00, ..., 1.5300e+01, 3.9690e+02, 4.9800e+00], [2.7310e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9690e+02, 9.1400e+00], ...])}`. В каждой строке 13 числовых значений. Ваша задача найти описание каждого из них. Так первое значение (`6.3200e-03` или `0.00632`) это уровень преступности на душу населения. Постройте дерево решений, предсказывающее стоимость жилья по имеющимся данным. Обратите внимание, стоимость жилья - величина непрерывная, и дерево классификации `DecisionTreeClassifier` нам не подойдёт. Для предсказания значения целевой функции нужно регрессионное дерево `DecisionTreeRegressor`. Переберите все деревья на данных `boston` по следующим параметрам: максимальная глубина - от 2 до 15 уровней. минимальное число проб для разделения - от 2 до 10 минимальное число проб в листе - от 1 до 10. Отобразите лучшие параметры и лучшее дерево.

Задание 26

Загрузите встроенный набор `boston`. Изучите остальные атрибуты классификатора. Создайте модель `RandomForestClassifier` с 15 деревьями и максимальной глубиной равной 5 уровням для данных `digits_load()`. Обучите полученную модель. Выполните предсказание. Посчитайте точность (`score`) для леса. Укажите три самых важных параметра для деревьев решений в созданном лесу, используя, например, атрибут `feature_importances_`.

Задание 27

Загрузите `log`-файл активности пользователей в ЭИОС вуза. Проведите первичный анализ. Сгруппируйте данные по пользователю. Выведите гистограмму по времени проводимому в системе. Постройте круговую диаграмму, отображающую затраты времени по разделам. Ответьте письменно, сколько времени в системе проводят чаще всего, какой раздел посещают чаще всего.

Задание 28

Загрузите встроенный набор `boston`. Набор данных `boston` представляет собой словарь с ключами `data`, `target`, `feature_names` и `DESCR`. Ключу `data` соответствует массив списков: `{'data': array([[6.3200e-03, 1.8000e+01, 2.3100e+00, ..., 1.5300e+01, 3.9690e+02, 4.9800e+00], [2.7310e-02, 0.0000e+00, 7.0700e+00, ..., 1.7800e+01, 3.9690e+02, 9.1400e+00], ...])}`. Используя алгоритм случайный лес (`random forest`) Создайте модель `RandomForestClassifier` с 20 деревьями и максимальной глубиной равной 5 уровням для данных. Обучите полученную модель. Выполните предсказание. Посчитайте точность (`score`) для леса.

Задание 29

Скачайте данные с сайта `kugle` (<https://www.kaggle.com/c/dogs-vs-cats/data>). Для скачивания необходимо пройти процедуру регистрации. Распакуйте архивы `train.zip` и `test.zip` с данными в каталоге со своим проектом. В каталоге `train` создайте две папки `cat` и `dog`. Разнесите изображения из каталога `train` по вновь созданным папкам. Вы можете сделать это вручную (0 баллов) или программно (1 балл) используя методы языка Python. Вам понадобится библиотека `os` и методы строк. Сгенерируйте датасет из изображений используя метод библиотеки `keras ImageDataGenerator`. С помощью метода `flow_from_directory` создайте генератор данных на основе изображений из каталога. Исследуйте полученный генератор. Выведите на экран, изучите форму. Проведите обратную операцию. Визуализируйте первые пять объектов из полученного набора данных. Расщепите имеющиеся данные на три набора (`train, test, validation`), либо

используйте данные из архива test, подготовив их соответствующим образом. Самостоятельно создайте и обучите нейронную сеть для различения кошек и собак (по примеру лабораторной 5, часть 2). Добейтесь точности модели на валидационной выборке не ниже 80%.

Задание 30

Реализуйте класс Neuron осуществляющий выполнение функции $y = x_1 * w_1 + x_2 * w_2 + \dots + x_n * w_n + b$ (1), где x – значения, поданные на вход, w - веса входов, а b - смещение. В качестве функции активации используйте сигмоиду. Убедитесь, что созданный вами нейрон работает. Для этого самостоятельно задайте веса и смещения и значения входов и проверьте ответ. Соберите из нейронов простейшую искусственную сеть, состоящую из трех слоев: входного, скрытого и выходного. Скрытый слой – это любой слой между входным (первым) слоем сети и выходным (последним). Скрытых слоев может быть несколько!

4. ОСНОВНЫЕ КРИТЕРИИ ОЦЕНИВАНИЯ ПРИ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Шкала оценивания уровня сформированности компетенций (по пятибалльной системе) экзамен

Формируемые уровни освоения компетенций	Критерии оценивания	Шкала оценивания
Высокий уровень	Изложено правильное понимание вопроса, четко и самостоятельно дан исчерпывающий ответ, содержание раскрыто полно, профессионально, грамотно. Обучающимся усвоена взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии. Отражает успешное и систематическое применение навыков и умений по данной дисциплине в соответствии с ФГОС.	отлично
Базовый уровень	Изложено правильное понимание вопроса, дано достаточно подробное описание предмета ответа, приведены и раскрыты в тезисной форме основные понятия, относящиеся к предмету ответа. Ответ отражает полное знание учебно-программного материала, систематический характер знаний по дисциплине, а также наличие базового уровня овладения практическими умениями и навыками по данной дисциплине в соответствии с ФГОС	хорошо
Пороговый уровень	Ответ отражает теоретические знания основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии. Данная оценка может быть выставлена обучающемуся, допустившему неточности в ответе, но обладающими необходимыми	удовлетворительно

	знаниями для их устранения под руководством преподавателя, отмечен начальный уровень овладения практическими умениями и навыками по данной дисциплине в соответствии с ФГОС	
Неудовлетворительный уровень	При ответе обучающегося обнаружено отсутствие знаний, умений и навыков и/или фрагментарные знания основного учебно-программного материала.	неудовлетворительно

Текущий контроль и промежуточная аттестация осуществляются в соответствии с «Положением о текущей и промежуточной аттестации обучающихся в Автономной некоммерческой организации «Образовательная организация высшего образования» «Университет экономики и управления».

Вид промежуточной аттестации – экзамен.

Форма проведения промежуточной аттестации – письменный экзамен.