

**Автономная некоммерческая организация профессионального образования  
«ПЕРМСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»  
(АНО ПО «ПГТК»)**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ  
ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ  
ПО ДИСЦИПЛИНЕ**

**МДК 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ В  
СИСТЕМАХ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА**

для специальности

**09.02.13 «Интеграция решений с применением технологий  
искусственного интеллекта»  
(код и наименование специальности)**

Квалификация выпускника

**Специалист по работе с искусственным интеллектом**

Форма обучения

**Очная**

Пермь 2026

Методические рекомендации по выполнению практических работ учебной дисциплины МДК 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ В СИСТЕМАХ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА составлен в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.13 «Интеграция решений с применением технологий искусственного интеллекта»

Данные методические рекомендации помогут организовать самостоятельную деятельность студентов на основе деятельного и компетентного подходов к обучению, что соответствует ФГОС СПО по специальности 09.02.13 «Интеграция решений с применением технологий искусственного интеллекта».

Программа предназначена для студентов и преподавателей АНО ПО «ПГТК».

Автор – составитель: Могильникова Н.С., старший преподаватель.

Методические рекомендации по выполнению практических работ предназначен для оценивания достижений запланированных результатов по дисциплине ОП.02 «ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ»

Методические рекомендации по выполнению практических работ представляет собой комплект материалов для проведения практических занятий (в форме практической подготовки) и осуществления контроля за выполнением работ.

Методические рекомендации по выполнению практических работ позволяет оценивать:

Код ОК, ПК	Уметь	Знать
ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части	актуальный профессиональный и социальный контекст, в котором приходится работать и жить
ОК 02 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности	определять этапы решения задачи, составлять план действия, реализовывать составленный план, определять необходимые ресурсы	структура плана для решения задач, алгоритмы выполнения работ в профессиональной и смежных областях
ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы	основные источники информации и ресурсы для решения задач и/или проблем в профессиональном и/или социальном контексте
ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.	владеть актуальными методами работы в профессиональной и смежных сферах	методы работы в профессиональной и смежных сферах
ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.	оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)	порядок оценки результатов решения задач
	определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации	профессиональной деятельности
	выделять наиболее значимое в перечне информации, структурировать получаемую информацию, оформлять результаты поиска	номенклатура информационных источников, применяемых в профессиональной деятельности
	оценивать практическую значимость результатов поиска	приемы структурирования информации
	применять средства информационных технологий для решения профессиональных задач	формат оформления результатов поиска информации
	использовать современное программное обеспечение в профессиональной деятельности	современные средства и устройства информатизации, порядок их применения
	использовать различные цифровые средства для решения профессиональных задач	программное обеспечение в профессиональной деятельности, в том числе цифровые средства
	грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке	психологические основы деятельности коллектива
	проявлять толерантность в рабочем коллективе	правила оформления документов
	Анализировать технические задания и выявлять требования к алгоритмам.	правила построения устных сообщений
		особенности социального и культурного контекста
		Основные методы и подходы к построению алгоритмов (типовые поисковые алгоритмы, жадные алгоритмы, динамическое программирование, рекурсивные подходы).
		Принципы эффективной обработки данных.

<p>ПК 1.3 Оформлять программный код в соответствии с техническим заданием.</p> <p>ПК 1.4 Использовать систему контроля версий программного кода с учетом обеспечения возможности организации групповой разработки.</p> <p>ПК 1.5 Выполнять отладку программных модулей с использованием специализированных программных средств.</p> <p>ПК 1.6 Выполнять тестирование программного кода.</p> <p>ПК 1.7 Составлять тестовые сценарии.</p>	<p>Применять методы алгоритмизации для решения задач программирования.</p> <p>Разрабатывать оптимальные алгоритмы для решения задач в области ИИ.</p> <p>Реализовывать программные модули на основе требований технического задания.</p> <p>Соблюдать при разработке принципы «чистого кода».</p> <p>Использовать стандартные библиотеки и фреймворки для ускорения разработки.</p> <p>Оформлять код в соответствии с принятыми стандартами и требованиями.</p> <p>Документировать разработанный программный код.</p> <p>Соблюдать соглашения о наименованиях переменных, функций и классов (например, PEP8 для Python).</p> <p>Работать с системами контроля версий для управления проектами.</p> <p>Организовывать совместную работу над проектом через ветки разработки и слияние изменений.</p> <p>Разрешать конфликты при слиянии кода.</p> <p>Использовать инструменты для отладки программного кода.</p> <p>Идентифицировать и исправлять ошибки в программе.</p> <p>Применять методы логирования для анализа выполнения программ.</p> <p>Проводить различные виды тестирования (юнит-тестирование, интеграционное тестирование).</p> <p>Выполнять настройки окружения и подготовку тестовых данных</p> <p>Фиксировать результаты выполнения тестов и подготавливать отчеты о результатах тестов.</p> <p>Определять уровень критичности дефектов.</p> <p>Разрабатывать автоматизированные тесты для тестирования модулей и/или отдельных функций</p> <p>Восстанавливать окружение и тесты после сбоя</p> <p>Проектировать тестовые сценарии на основе тестовых планов.</p>	<p>Языки программирования, применяемые для разработки алгоритмов.</p> <p>Принципы модульного программирования.</p> <p>Языки программирования для разработки модулей.</p> <p>Стандартные фреймворки и библиотеки для работы с ИИ.</p> <p>Основные принципы чистого кода (Clean Code).</p> <p>Стандарты и практики документирования программного обеспечения.</p> <p>Инструменты для автоматической проверки качества кода (например, PyLint, ESLint).</p> <p>Принципы работы распределенных систем контроля версий.</p> <p>Основные команды и операции в системах контроля версий (например: commit, pull, push, merge).</p> <p>Методы разрешения конфликтов в ходе групповой разработки.</p> <p>Принципы работы отладчиков и логирования.</p> <p>Способы выявления ошибок в программе (отладка по шагам, точки останова).</p> <p>Инструменты для отладки кода (например, PyCharm, Visual Studio Debugger).</p> <p>Техники выполнения тестовых прогонов.</p> <p>Инструменты и среды выполнения тестирования</p> <p>Языки разработки автоматизированных тестов</p> <p>Инструменты для тестирования программного кода.</p> <p>Правила выполнения отчетов о тестировании</p> <p>Цели, задачи и виды тестирования. Понятие стратегии тестирования.</p> <p>Жизненный цикл дефекта.</p> <p>Основы тест-дизайна: тестовый сценарий, тестовый пакет, чек-лист, основные шаблоны.</p>
---	---	---

	Разрабатывать тестовые пакеты и задания на выполнение тестирования. Использовать шаблоны для написания тест-кейсов. Оценивать риски при отборе тестов для регрессионного тестирования. Оценивать тесты на соответствие целям тестирования.	Основные инструменты проектирования тестов. Методы и подходы к написанию тестов (Test-Driven Development, Behavior-Driven Development).
--	---	--

## Перечень заданий для практических работ

### Тема 1.1. Введение в искусственный интеллект и его направления

#### Практическое занятие №1: Анализ примеров успешных решений на основе ИИ

##### Цель занятия

Изучить и проанализировать успешные кейсы внедрения ИИ в различных отраслях.

##### План занятия

##### Практическая работа (3 часа)

- **Групповая работа:** анализ кейсов по направлениям:
  - Здравоохранение (диагностика заболеваний)
  - Финансы (анализ рисков)
  - Производство (оптимизация процессов)
  - Розничная торговля (персонализация предложений)
- **Исследование кейсов:**
  - Описание проблемы
  - Решение с помощью ИИ
  - Достигнутые результаты
  - Экономический эффект

##### Часть 3. Презентация и обсуждение (1 час)

- Защита групповых проектов
- Сравнительный анализ решений
- Формулировка выводов

##### Форма отчетности

- Презентация кейса
- Письменный отчет с анализом
- Выводы по эффективности решений

#### Практическое занятие №2: Создание базовой модели ИИ для классификации данных (5 часов)

##### Цель занятия

Получить практические навыки разработки простой модели машинного обучения.

##### План занятия

##### Часть 1. Подготовка (1 час)

- Установка необходимого ПО (Python, библиотеки ML)
- Выбор набора данных
- Создание рабочей среды

##### Часть 2. Разработка модели (3 часа)

- **Этап 1:** предобработка данных
  - Загрузка и очистка данных
  - Визуализация
  - Разделение на выборки
- **Этап 2:** построение модели
  - Выбор алгоритма классификации

- Настройка параметров
- Обучение модели
- **Этап 3:** оценка результатов
  - Тестирование
  - Анализ ошибок
  - Оптимизация

### **Часть 3. Защита работы (1 час)**

- Демонстрация результатов
- Обсуждение метрик качества
- Формулировка выводов

### **Необходимые материалы**

- Наборы данных
- Инструкции по установке ПО
- Шаблоны кода
- Методические указания

### **Форма отчетности**

- Код реализации модели
- Результаты тестирования
- Отчет с анализом
- Презентация решения

## **Тема 1.2. Методы сбора и предобработки данных**

**Цель:** приобретение умений и навыков при решении комбинаторных уравнений.

**Практическое занятие №3.** Сбор данных с использованием веб-скрапинга и API

Цель занятия: освоить методы автоматизированного извлечения данных из веб-ресурсов и сервисов посредством скрапинга и обращения к API.

### **План занятий:**

Практическое задание по веб-скрапингу

Разделите студентов на пары. Каждой паре дается ссылка на сайт, содержащий нужную информацию (например, список книг с оценками читателей). Задача каждой команды — написать скрипт, собирающий данные с сайта и сохраняющий их в удобочитаемом формате (CSV или JSON).

Работа с API

После успешной сдачи первого этапа преподаватель демонстрирует работу с публичными API (например, GitHub API или OpenWeatherMap API). Каждому студенту выдается индивидуальное задание по извлечению конкретных данных с помощью API.

Совместная сессия вопросов и ответов

Все участники объединяются вместе, задают вопросы друг другу и учителю относительно проблем, возникших в ходе работы над заданиями.

Заключительная презентация итогов

Каждый участник показывает собранный набор данных и делится впечатлениями о работе с разными источниками данных.

**Практическое занятие №4.** Предобработка данных для машинного обучения: очистка, нормализация, кодирование

Цель занятия: научиться правильно готовить данные для дальнейшего анализа и моделирования в рамках машинного обучения.

**План занятий:**

Краткий теоретический блок

Преподаватель напоминает ключевые концепции предобработки данных: заполнение пропусков, устранение выбросов, масштабирование признаков, обработка категориальных переменных.

Индивидуальная практика по очистке данных

Каждому участнику предлагается набор сырых данных (например, набор клиентских заявок банка или показатели производительности сотрудников). Задания включают выявление пропущенных значений, замену неверных записей, нормализацию числовых признаков.

Предобработка категориальных данных

Участники изучают способы преобразования категориальных признаков в числовую форму (one-hot encoding, label encoding). Выполняются упражнения по применению каждого метода.

Автоматизация процесса предобработки

Используя библиотеку Python pandas, студенты создают автоматические скрипты для обработки типичных ошибок и повторяющихся операций очистки данных.

Подведение итогов и отчетность

Студенты представляют обработанные данные и рассказывают о методах, применённых ими для достижения лучших результатов.

Критерий оценки:

4 - 5 заданий – оценка 3

6 - заданий – оценка 4

7 – заданий – оценка 5

**Тема 1.3. Основы алгоритмов машинного обучения**

**Практическое занятие №5.** Реализация линейной регрессии на реальных данных

Цель занятия: Освоение методов реализации и интерпретации результатов линейной регрессии на практике.

**План занятия:**

**Выбор и подготовка набора данных**

1. Студентам предоставляется реальный набор данных (например, зависимость цены

квартиры от площади, количества комнат, местоположения и др.). Данные предварительно очищаются преподавателем, однако остаются небольшие недостатки, которые предстоит устранить самим учащимся.

### **Разделение данных на тренировочную и тестовую выборки**

2. Учащиеся самостоятельно выполняют разделение данных на train/test выборки, выбирают пропорции разделения (обычно 80%/20%).

### **Реализация простой линейной регрессии**

3. Индивидуально или парами учащиеся реализуют простейшую одномерную линейную регрессию (например, цена зависит только от общей площади жилья):
  - строят график зависимости,
  - оценивают коэффициенты уравнения,
  - интерпретируют полученные значения коэффициентов.

### **Многомерная линейная регрессия**

4. Далее учащиеся переходят к построению многофакторной модели, включающей несколько факторов (например, площадь, количество комнат, этажность, район города):
  - учатся оценивать значимость отдельных факторов,
  - проверяют мультиколлинеарность и устраняют избыточные признаки,
  - сравнивают качество однофакторной и многофакторной моделей.

### **Интерпретация результатов и визуализация**

5. Завершающим этапом является представление полученных результатов группе:
  - демонстрация графиков зависимостей,
  - пояснение значимости найденных коэффициентов,
  - вывод о качестве полученной модели.

## **Практическое занятие №6. Применение кластеризации для сегментации данных**

**Цель занятия:** Овладеть основными алгоритмами кластеризации и применять их для сегментирования реальных данных.

### **План занятия:**

1. **Знакомство с концепциями кластеризации**
2. Преподаватель знакомит учащихся с базовыми понятиями кластерного анализа: цели, виды алгоритмов (k-means, иерархическая кластеризация, DBSCAN), критерии выбора числа кластеров (метрика silhouette score, elbow method).
3. **Выбор и подготовка данных**
4. Предоставляется набор реальных данных (например, характеристики клиентов магазина: возраст, доход, частота покупок, сумма среднего чека).



## 5. Нормализация и стандартизация данных

6. Ученики нормализуют и стандартизируют данные, поскольку большинство алгоритмов чувствительны к масштабу признаков.

## 7. Применение k-means-кластеризации

8. Учащиеся индивидуально или небольшими группами применяют алгоритм k-means:

- определяют оптимальное число кластеров методом локтя («elbow method») и силуэт-анализа,
- запускают алгоритм и получают сегменты,
- визуально отображают получившиеся кластеры.

## 9. Использование альтернативных методов

10. Для сравнения эффективности проводится эксперимент с другим алгоритмом (например, DBSCAN или иерархическим кластером):

- ученики настраивают параметры алгоритма,
- сравнивают полученный результат с предыдущими результатами.

## 11. Представление результатов и выводы

12. Заключительный этап включает презентацию результатов групповым коллегам:

- демонстрируются графики распределения объектов по кластерам,
- объясняются различия в поведении разных алгоритмов,
- даются рекомендации по выбору оптимального подхода для дальнейших исследований.

Таким образом, оба занятия обеспечивают полное погружение учеников в процесс самостоятельного исследования реальных данных, позволяют закрепить знания теории и развить навыки самостоятельной постановки экспериментов и интерпретации результатов.

### Тема 1.4. Оценка качества моделей и улучшение алгоритмов

#### Практическое занятие №7. Оценка качества модели с использованием ROC-кривой и F-меры

**Цель занятия:** Научиться эффективно оценивать качество бинарной классификации с помощью кривых ROC-AUC и показателя F-меры.

#### План занятия:

1. **Введение в оценку качества моделей**
2. Преподаватель объясняет базовые понятия: точность (accuracy), чувствительность (recall), специфичность (specificity), precision, F-меру, AUC-ROC.
3. **Изучение ROC-кривой** Учащимся объясняется принцип построения ROC-кривой, расчет показателей AUC (Area Under Curve), важность правильного понимания

порога вероятности принятия решения.

4. **Расчет F-меры**

5. Демонстрация расчета F-меры, показываются случаи, когда предпочтительнее использовать F-меру вместо стандартных метрик точности.

6. **Практическое задание:**

7. Участникам выдаётся заранее подготовленный набор данных (например, медицинские диагнозы или кредитоспособность заемщиков). Необходимо реализовать следующую последовательность шагов:
- построить классификационную модель (логистическая регрессия, случайный лес);
  - рассчитать и построить ROC-кривую, оценить показатель AUC;
  - рассчитать Precision, Recall и F-меру;
  - сравнить эффективность выбранной модели с эталонной моделью (baseline model);
  - сделать выводы о качестве модели и выбрать оптимальный порог принятия решения.
8. **Отчётность и защита результатов**
9. Презентация результатов каждому участнику или команде, обсуждение преимуществ и недостатков выбранных подходов.

## **Практическое занятие №8. Настройка гиперпараметров модели с использованием GridSearchCV**

**Цель занятия:** Освоить эффективные подходы настройки гиперпараметров моделей машинного обучения с применением инструмента GridSearchCV.

**План занятия:**

1. **Что такое гиперпараметры и зачем их оптимизировать**
2. Преподаватель разъясняет понятие гиперпараметров (например, глубина деревьев, регуляризаторы, learning rate), причины необходимости оптимизации.
3. **Методы подбора гиперпараметров**
4. Рассматриваются ручная настройка, grid search, random search, bayesian optimization.
5. **GridSearchCV: теория и практика**
6. Подробно рассматривается библиотека sklearn.model\_selection.GridSearchCV, принципы её работы, синтаксис и особенности применения.
7. **Практическое задание:**

8. Работа выполняется индивидуально либо в небольших группах. Учащимся предоставляются реальные данные (например, данные о клиентах банка или медицинских пациентах). Требуется:

- выбрать подходящую модель (например, Random Forest, XGBoost, SVM);
- сформулировать гипотезу о влиянии ключевых гиперпараметров на качество модели;
- настроить гиперпараметры с помощью GridSearchCV;
- проанализировать влияние изменения гиперпараметров на производительность модели;
- представить наилучшие комбинации гиперпараметров и соответствующие результаты.

#### 9. Обсуждение и отчётность

10. Выступления команд или индивидуальных участников с представлением результатов, обсуждением выводов и рекомендациями по дальнейшему улучшению моделей.

Данные занятия позволят участникам глубоко разобраться в оценке качества моделей и настройке их параметров, что существенно повысит уровень компетенций в анализе данных и разработке эффективных моделей машинного обучения.

#### **При выполнении практической работы и контрольной работы:**

Содержание и объем материала, подлежащего проверке в контрольной работе, определяется программой. При проверке усвоения материала выявляется полнота, прочность усвоения учащимися теории и умение применять ее на практике в знакомых и незнакомых ситуациях.

Отметка зависит также от наличия и характера погрешностей, допущенных учащимися.

- *грубая ошибка* - полностью искажено смысловое значение понятия, определения;
- *погрешность* отражает неточные формулировки, свидетельствующие о нечетком представлении рассматриваемого объекта;
- *неДочет* - неправильное представление об объекте, не влияющего кардинально на знания определенные программой обучения;
- *мелкие погрешности* - неточности в устной и письменной речи, не искажающие смысла ответа или решения, случайные описки и т.п.

Эталоном, относительно которого оцениваются знания учащихся, является обязательный минимум содержания информатики и информационно-коммуникационных технологий.

Исходя из норм (четырёхбалльной системы), заложенных во всех предметных областях выставляете отметка:

«5» ставится при выполнении всех заданий полностью или при наличии 1 -2 мелких погрешностей;

«4» ставится при наличии 1-2 недочетов или одной ошибки:

«3» ставится при выполнении  $\frac{2}{3}$  от объема предложенных заданий;

«2» ставится, если допущены существенные ошибки, показавшие, что учащийся не владеет обязательными умениями поданной теме в полной мере (незнание основного программного материала)

## 2.2. Перечень вопросов и заданий для промежуточной аттестации

Форма: дифференцированный зачет

### Список вопросов к дифференцированному зачету

#### МДК 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ В СИСТЕМАХ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

1. Основные компоненты системы искусственного интеллекта. Их назначение и взаимосвязь.
2. Понятие и типы программного модуля в искусственном интеллекте. Отличительные черты модульного проектирования.
3. Современные языки программирования, используемые в сфере искусственного интеллекта. Преимущества и ограничения каждого из них.
4. Структура программы на Python для решения задач искусственного интеллекта. Особенности и шаблоны проектирования.
5. Алгоритмы машинного обучения: классификация, регрессия, кластеризация. Примеры реализации.
6. Что такое пакет NumPy и почему он важен для работы с массивами данных в AI-приложениях?
7. Библиотека Pandas: структура DataFrame, операции фильтрации, агрегации и группировки данных.
8. Процесс предварительной обработки данных (Data Preprocessing): очистка, нормализация, уменьшение размерности.
9. Выбор и обучение модели: кросс-валидация, гипер-параметры, регуляризация.
10. Инструменты визуализации данных: Matplotlib, Seaborn. Зачем они нужны и как используются?
11. Что такое TensorFlow и PyTorch? Назначение и область применения этих фреймворков.
12. Глубокое обучение: архитектура нейронных сетей, прямые и обратные проходы, стохастический градиентный спуск.
13. Типичные архитектуры искусственных нейронных сетей: Perceptron, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs).
14. Метод Grid Search CV: цель, алгоритм работы, плюсы и минусы.
15. Техники обработки естественного языка (Natural Language Processing, NLP): токенизация, векторизация текста, Word Embeddings.
16. Роль предобученных моделей (Transfer Learning) в современных приложениях искусственного интеллекта.
17. Машинное зрение: методы обнаружения и выделения объектов на изображениях.
18. Нейроэволюционные подходы: генетические алгоритмы, эволюционная стратегия, применение в проектировании нейронных сетей.
19. Искусственный интеллект в робототехнике: применение, современные тенденции и перспективы.
20. Этические аспекты использования искусственного интеллекта. Проблемы конфиденциальности и предвзятости.
21. Архитектуры программных компонентов в распределённом обучении: Spark, Hadoop, Apache Beam.
22. Современные облачные сервисы для вычислений в области искусственного интеллекта: Google Cloud Platform, Amazon Web Services, Microsoft Azure.
23. Оптимизация производительности программных модулей: параллельные вычисления, GPU ускорение, кэширование.
24. Тестирование и отладка программных модулей для искусственного интеллекта: unit-тесты, интеграционное тестирование, нагрузочное тестирование.
25. Безопасность и надёжность программного обеспечения в системах искусственного интеллекта: уязвимости, меры защиты.
26. Принцип работы алгоритма Gradient Boosting Machines (GBMs) и его отличие от Random Forests.
27. Байесовский подход в машинном обучении: вероятностные модели, байесовская логика, сравнение с частотным подходом.
28. Автоматизированное создание признаков (Feature Engineering): какие техники применяются и зачем это важно?
29. Что такое AutoML? Как используются платформы для автоматического машинного обучения?
30. Основные проблемы современного компьютерного зрения: ограничение точности, вычислительные затраты, масштабируемость.
31. Deep Reinforcement Learning: принцип работы, сценарии применения, сравнительный анализ DQN и Policy Gradients.
32. Обзор популярных библиотек и инструментов для работы с большими данными (Apache Kafka,

Cassandra, Redis).

33. Специфические задачи обработки больших объемов данных (Big Data) и стратегии их решения в условиях ограниченности ресурсов.
34. Способы борьбы с переобучением (Overfitting): Regularization, Dropout, Early Stopping, Bagging, Boosting.
35. Интерфейсы взаимодействия пользователей с системами искусственного интеллекта: голосовые ассистенты, чат-боты, интерфейсы умных устройств.
36. Контекстуальные факторы успеха внедрений AI-проектов: организационный, экономический, технологический аспекты.
37. Почему важны объяснимые модели искусственного интеллекта (Explainable AI)? Как решить проблему непрозрачности чёрных ящиков?
38. Перспективы интеграции квантовых компьютеров и искусственного интеллекта: потенциальные выгоды и препятствия.
39. Новые подходы в области обучения с учителем (Supervised Learning): semi-supervised, self-supervised, one-shot learning.
40. Инновационные приложения искусственного интеллекта в медицине, биологии, образовании, транспорте.
41. Ограничения и риски нейроморфных вычислений: аппаратные требования, энергопотребление, надежность.
42. Принципы эффективной организации хранения и управления наборами данных для искусственного интеллекта.
43. Современные стандарты тестирования и верификации программных продуктов в области искусственного интеллекта.
44. Важнейшие метрики оценки качества моделей классификации и регрессии: Accuracy, Precision, Recall, MSE, MAE, MAPE.
45. Какой вклад вносят Graph Neural Networks (сети на графах) в развитие систем искусственного интеллекта?

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

<b>№ п.п.</b>	<b>Содержание изменения</b>	<b>Дата, номер протокола заседания кафедры, подпись зав.кафедрой</b>
1	2	3
1		
2		
3		
4		