МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению практических работ
по дисциплине «Искусственный интеллект и машинное обучение»
для студентов направления подготовки
09.03.01 Информатика и вычислительная техника
Направленность (профиль)
«Программное обеспечение средств вычислительной техники и автоматизированных систем»

Ставрополь 2026

Содержание

ВВЕДЕНИЕ

1. Цель и задачи освоения дисциплины (модуля)

Цель дисциплины – формирование, согласно образовательной программы, компетенций ПК-1, ПК-2 обучающегося.

Для достижения этой цели в процессе изучения дисциплины решаются следующие задачи: ознакомление с основными понятиями и типами задач машинного обучения; изучение методов подготовки данных, включая очистку, нормализацию и инженерное преобразование признаков; овладение базовыми моделями обучения с учителем, такими как линейная и логистическая регрессия; освоение стохастического градиентного спуска как метода оптимизации моделей; формирование представлений о метриках оценки качества классификации и интерпретации результатов моделей; изучение методов борьбы с переобучением, включая регуляризацию и кросс-валидацию; освоение алгоритмов классификации на основе метода ближайших соседей и метода опорных векторов; изучение алгоритмов построения деревьев решений и ансамблевых моделей, таких как случайный лес и градиентный бустинг; знакомство с методами обучения без учителя, включая алгоритмы кластеризации; овладение методами уменьшения размерности и визуализации данных, такими как PCA и t-SNE; развитие практических навыков настройки моделей с использованием подбора гиперпараметров; формирование умений сравнивать и интерпретировать результаты различных моделей на основе объективных метрик.

2. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина относится к части, формируемой участниками образовательных отношений.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ПК-1. Способен разрабатывать требования и проектировать программное обеспечение	ИД-2$_{ПК-1}$ проводит анализ исполнения требований; вырабатывает варианты реализации требований; проводит оценку и обоснование рекомендуемых решений; выбирает средства реализации требований к программному обеспечению; использует существующие типовые решения и шаблоны проектирования программного обеспечения; разрабатывает технические спецификации на программные компоненты и их взаимодействие.	Проведение анализа исполнения требований; выработка вариантов реализации требований; проведение оценки и обоснование рекомендуемых решений; выбор средств реализации требований к программному обеспечению; использование существующие типовых решений и шаблонов проектирования программного обеспечения; разработка технических

		спецификаций на программные компоненты и их взаимодействие.
ПК-2. Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности	ИД-1$_{ПК-2}$ реализует методы планирования проектных работ; применяет методы классического системного анализа; использует теорию управления бизнес-процессами, методы концептуального проектирования, методы оценки качества программных систем; осуществляет решение задач оптимизации при разработке бизнес-требований к системе.	Понимание методов классического системного анализа; понимание теории управления бизнес-процессами; понимание методов концептуального проектирования; понимание методов оценки качества программных систем.
	ИД-2$_{ПК-2}$ разрабатывает технико-экономическое обоснование; разрабатывает техническое задание на систему; разрабатывает требования к подсистемам системы и осуществляет контроль их качества; организовывает оценку соответствия требованиям существующих систем и их аналогов; выполняет сопровождение приемочных испытаний и ввод в эксплуатацию системы; обрабатывает запросы на изменение требований к системе.	Способность: разработать технико-экономическое обоснование; разработать техническое задание на систему; разработать требования к подсистемам системы и осуществить контроль их качества; организовать оценку соответствия требованиям существующих систем и их аналогов; выполнить сопровождение приемочных испытаний и ввод в эксплуатацию системы; обработать запросы на изменение требований к системе.
	ИД-3$_{ПК-2}$ составляет графики контрольных мероприятий; разрабатывает бизнес-требования к системе; способен определять ключевые свойства и ограничения системы; выделяет подсистемы системы	Способность составлять графики контрольных мероприятий; разрабатывать бизнес-требования к системе; способен определять ключевые свойства и ограничения системы; выделяет подсистемы системы
	ИД-4$_{ПК-2}$ Осуществляет решение задач оптимизации при разработке бизнес-требований к системе	Способность решать оптимизационные задачи при проектировании систем машинного обучения.

	ИД-12$_{ПК-2}$ Разрабатывает программный код на языках программирования высокого уровня	Способен разрабатывать системы машинного обучения на современных языках высокого уровня.
	ИД-15$_{ПК-2}$ Осуществляет сбор, обработку и анализ результатов оценки готовых систем на соответствие требований	Осуществление сбора обработки и анализа результатов готовых систем машинного обучения на соответствие требованиям.

4. Объем учебной дисциплины (модуля) и формы контроля *

Объем занятий: всего: 3 з.е. 108 акад.ч.	ОФО, В акад. часах
Контактная работа:	64.00/32.00
Лекции/из них практическая подготовка	32.00/0
Лабораторных работ/из них практическая подготовка	32.00/32.00
Практических занятий/из них практическая подготовка	
Самостоятельная работа	44.00
Формы контроля	
Зачет с оценкой	да

5. Важность практической работы студентов

Практическая работа студентов является центральным элементом освоения дисциплины *«Искусственный интеллект и машинное обучение»*. В условиях стремительного развития технологий и высокой прикладной направленности этой области, теоретические знания без практического применения теряют актуальность и глубину.
Основная цель практических занятий — не только закрепление теоретического материала, но и формирование у студентов реальных навыков работы с алгоритмами, библиотеками и инструментами анализа данных. Практика позволяет студентам:

- понять, как теория реализуется в коде и как работают алгоритмы на конкретных примерах;
- научиться работать с современными библиотеками Python, такими как Scikit-learn, Pandas и Matplotlib;
- отрабатывать полный цикл построения модели: от загрузки и очистки данных до оценки и интерпретации результатов;
- выявлять и устранять ошибки, возникающие при реализации моделей;
- сравнивать различные методы машинного обучения и выбирать наиболее подходящие под задачу;
- развивать аналитическое мышление, учиться интерпретировать метрики и визуализировать поведение модели.

Практическая работа также формирует важные компетенции, востребованные в профессиональной среде: умение самостоятельно находить решения, работать с открытыми данными, настраивать параметры моделей и оценивать их эффективность в реальных условиях. Регулярное выполнение практических заданий повышает мотивацию студентов, формирует уверенность в собственных силах и готовит к применению машинного обучения в реальных проектах и научных исследованиях.

ПРАКТИЧЕСКАЯ РАБОТА №1

Тема: Основы работы с Jupyter Notebook, JupyterLab и Google Colab

Цель работы:

Познакомиться с основами организации и ведения интерактивных вычислений в Jupyter Notebook, JupyterLab и Google Colab. Овладеть базовыми приёмами ввода, форматирования и выполнения кода и текстовых ячеек, а также загрузки и сохранения рабочих файлов.

Теоретическое обоснование:

Jupyter Notebook и JupyterLab представляют собой интерактивные среды для разработки и анализа данных, широко используемые в задачах машинного обучения и искусственного интеллекта. Эти среды поддерживают язык Python и позволяют совмещать программный код, пояснительный текст, визуализации и выводы в одном документе. JupyterLab является более современной и модульной средой, расширяющей возможности классического Jupyter Notebook.

Google Colab — это облачная платформа, предоставляющая пользователям бесплатный доступ к ноутбукам, которые можно запускать на удалённых серверах Google. Colab особенно полезен при отсутствии локальной среды и позволяет использовать вычислительные ресурсы, включая GPU и TPU.

Работа в интерактивных средах позволяет эффективно строить и отлаживать модели, документировать ход эксперимента, делиться результатами и сохранять их в едином файле. Студенты должны уметь различать типы ячеек (кодовые и текстовые), форматировать текст с помощью Markdown, вставлять формулы на языке LaTeX, а также управлять загрузкой данных и подключением библиотек.

Знание и уверенное владение Jupyter Notebook, JupyterLab и Google Colab является основой для дальнейшего успешного освоения дисциплины и выполнения лабораторных, проектных и научных задач.

Аппаратура и материалы:

Компьютер или ноутбук с доступом в интернет
Браузер Google Chrome или Mozilla Firefox
Установленная среда JupyterLab или Jupyter Notebook (через Anaconda или pip)
Учётная запись Google для доступа к Google Colab и Google Диску
Пример CSV-файла (предоставляется преподавателем или берётся из `sklearn.datasets`)
Программное обеспечение: Python 3.8+, библиотеки `numpy`, `pandas`, `matplotlib`

Типовые задачи:

1. Создайте Jupyter Notebook с кратким описанием цели будущего проекта и вставьте примеры кода, демонстрирующие выполнение арифметических операций.
2. Откройте Google Colab, создайте новый ноутбук, подключите его к вашему Google Диску и загрузите CSV-файл, затем выведите первые строки таблицы.
3. В JupyterLab создайте файл, содержащий текстовую ячейку с заголовком, маркированным списком и формулой на языке LaTeX, и кодовую ячейку, реализующую простую функцию.

Контрольные вопросы:

1. Чем Jupyter Notebook отличается от JupyterLab?
2. Какие преимущества предоставляет Google Colab по сравнению с локальными средами?
3. Как сохранить ноутбук на локальном компьютере и в облаке?
4. Каковы основные типы ячеек в Jupyter?
5. Как вставлять формулы и форматированный текст в ячейку Markdown?
6. Каким образом осуществляется подключение к Google Диску в Colab?
7. Какие команды используются для установки и подключения библиотек?
8. Что такое "runtime" в Google Colab?
9. Как загрузить данные в ноутбук и просмотреть их содержимое?
10. Какие форматы экспорта доступны для Jupyter Notebook?

ПРАКТИЧЕСКАЯ РАБОТА №2

Тема: Основы работы с пакетом NumPy

Цель работы:

Освоить базовые операции библиотеки NumPy, научиться создавать, индексировать и модифицировать многомерные массивы, выполнять векторные и матричные вычисления, а также использовать встроенные функции для анализа и обработки числовых данных.

Теоретическое обоснование:

Библиотека NumPy является фундаментальной для численных вычислений в Python и служит основой для большинства инструментов, используемых в машинном обучении и анализе данных. Основным объектом NumPy является многомерный массив `ndarray`, который обеспечивает высокую производительность за счёт хранения данных в компактной форме и возможности выполнять операции над массивами без использования циклов.

Работа с массивами позволяет эффективно представлять выборки данных, выполнять линейную алгебру, статистические операции, нормализацию и другие преобразования, необходимые на этапе предобработки данных. Одной из ключевых особенностей NumPy является возможность проводить операции поэлементно и векторизованно, что значительно ускоряет обработку больших объёмов информации.

NumPy также включает в себя набор инструментов для генерации случайных чисел, преобразования формы массивов, объединения и разбиения данных, а также выполнения логических операций и фильтрации. Глубокое понимание возможностей этой библиотеки необходимо для успешного освоения таких инструментов, как pandas, scikit-learn и другие библиотеки машинного обучения.

Аппаратура и материалы:

Компьютер или ноутбук с доступом в интернет
Локальная среда Jupyter Notebook или Google Colab
Установленная библиотека `numpy` (входит в состав Anaconda или устанавливается через pip)
Примеры массивов и задания, подготовленные преподавателем или сформированные

студентом самостоятельно
Дополнительные материалы: документация https://numpy.org/doc

Типовые задачи:

1. Создайте одномерный и двумерный массивы с помощью `np.array()`, выведите их форму, размер и тип данных.
2. Сгенерируйте массив из случайных чисел с использованием `np.random`, найдите среднее, минимум и максимум.
3. Выполните матричное умножение двух совместимых массивов, используя как `@`, так и `np.dot()`.

Контрольные вопросы:

1. Что такое объект `ndarray` в библиотеке NumPy?
2. Чем отличается поэлементное умножение от матричного?
3. Как узнать размерность массива и количество его элементов?
4. Как создавать массивы, заполненные нулями, единицами и случайными числами?
5. Как индексируются и срезаются массивы NumPy?
6. Что такое векторизация и почему она важна?
7. Как изменить форму массива (reshape) и когда это требуется?
8. Какие функции используются для объединения и разбиения массивов?
9. Как выполнять логические фильтрации и маскирование данных в массиве?
10. Какие преимущества использования NumPy по сравнению со стандартными списками Python?

ПРАКТИЧЕСКАЯ РАБОТ №3

Тема: Основы работы с пакетом Matplotlib

Цель работы:
Освоить основные средства визуализации данных с помощью библиотеки Matplotlib, научиться создавать линейные графики, гистограммы, диаграммы рассеяния и настраивать элементы графиков для наглядного представления информации.

Теоретическое обоснование:
Библиотека Matplotlib является стандартным инструментом для визуализации данных в Python и активно используется в задачах анализа данных и машинного обучения. Основной модуль библиотеки — `pyplot` — предоставляет интерфейс, аналогичный графическим возможностям MATLAB, и позволяет быстро строить графики различного типа.

Визуализация является важнейшим этапом при анализе данных. Графики позволяют выявить тенденции, выбросы, закономерности и ошибки в данных. Они также служат средством представления результатов построения моделей, сравнения предсказаний и интерпретации зависимостей между признаками.

С помощью Matplotlib можно строить линейные графики, гистограммы, столбчатые диаграммы, круговые диаграммы, графики рассеяния, настраивать заголовки, подписи осей, легенды и стили линий. Кроме того, библиотека позволяет сохранять графики в различных форматах и интегрировать их в отчёты и презентации.

Навыки визуализации особенно важны для коммуникации результатов в научной, учебной и прикладной деятельности.

Аппаратура и материалы:

Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленная библиотека `matplotlib` (в составе Anaconda или устанавливается через pip)
Дополнительные библиотеки: `numpy`, `pandas` для подготовки данных
Примеры данных: вручную созданные или из `sklearn.datasets`

Типовые задачи:

1. Постройте линейный график функции `y = sin(x)` на интервале от -2π до 2π. Настройте цвет, стиль линии, подписи осей и заголовок.
2. Постройте гистограмму распределения случайных чисел, сгенерированных функцией `np.random.randn()`. Настройте количество интервалов.
3. Постройте диаграмму рассеяния (`scatter plot`) для случайных точек и добавьте цветовую шкалу (`color map`) в зависимости от третьей переменной.

Контрольные вопросы:

1. Какова роль библиотеки Matplotlib в анализе данных?
2. В чём разница между линейным графиком и диаграммой рассеяния?
3. Какие функции модуля `pyplot` используются для построения графиков?
4. Как изменить цвет и стиль линии на графике?
5. Как задать заголовок и подписи к осям графика?
6. Как отображаются несколько графиков на одном рисунке?
7. Как строится гистограмма и что означает параметр `bins`?
8. Что такое `figure` и `subplot` в контексте Matplotlib?
9. Как сохранить график в файл?
10. Какие типы графиков наиболее подходят для визуализации взаимосвязей между признаками?

ПРАКТИЧЕСКАЯ РАБОТА №4

Тема: Введение в pandas: объект Series

Цель работы:

Изучить структуру и свойства объекта Series библиотеки pandas, освоить основные операции создания, индексирования, фильтрации и применения агрегатных функций к данным.

Теоретическое обоснование:

Библиотека pandas является одной из ключевых библиотек Python для обработки табличных и структурированных данных. Она предоставляет два основных типа объектов — Series и DataFrame. Объект Series представляет собой одномерный массив с метками (индексами), что делает его удобным для хранения и анализа упорядоченных последовательностей данных.

Series может содержать значения различных типов, в том числе числовые, строковые и логические. Он тесно интегрирован с библиотекой NumPy и поддерживает векторные операции, что делает его эффективным инструментом для выполнения типичных задач анализа данных.

Работа с Series включает создание вручную и на основе других структур данных (списки, словари, массивы NumPy), доступ к элементам по индексу или маске, применение арифметических операций, использование функций агрегирования, сортировку, а также проверку наличия пропущенных значений. Понимание устройства и поведения объекта Series является необходимым шагом перед изучением структуры DataFrame, более сложного и мощного инструмента для работы с табличными данными.

Аппаратура и материалы:

Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленная библиотека `pandas` (в составе Anaconda или через pip)
Дополнительно: библиотеки `numpy` и `matplotlib`
Исходные данные — вручную заданные или сгенерированные пользователем списки, словари, массивы

Типовые задачи:

1. Создайте объект Series из списка, словаря и массива NumPy. Отобразите значения, индексы и тип данных.
2. Примените методы `head()`, `tail()`, `describe()` и арифметические операции к Series.
3. Выполните логическую фильтрацию Series: выберите элементы, удовлетворяющие заданному условию.

Контрольные вопросы:

1. Что такое объект Series и в чём его отличие от обычного списка Python?
2. Какие способы создания Series вы знаете?
3. Как получить доступ к элементу Series по индексу и по значению?
4. Как изменяется индекс при создании Series из словаря?
5. Что происходит при выполнении арифметических операций над Series?
6. Какие методы используются для получения статистики и описания данных?
7. Как осуществляется фильтрация значений в Series?
8. Что такое маскирование и логическая индексация?
9. Как проверить наличие пропущенных значений в Series?
10. Какие преимущества работы с Series по сравнению со списками Python?

ПРАКТИЧЕСКАЯ РАБОТА №5

Тема: Введение в pandas: объект DataFrame

Цель работы:

Овладеть базовыми приёмами создания и использования объекта DataFrame библиотеки pandas, научиться выполнять операции по просмотру, выборке, фильтрации, сортировке и агрегации данных.

Теоретическое обоснование:

Объект DataFrame — основной структурный элемент библиотеки pandas — представляет собой двумерную таблицу с именованными строками (индексами) и столбцами. Он предназначен для представления и анализа табличных данных, аналогично таблицам в реляционных базах данных или электронных таблицах.

Создание DataFrame возможно из различных источников: словарей, списков, массивов NumPy, CSV-файлов и других форматов. Каждому столбцу в DataFrame

соответствует объект Series, что позволяет применять ко всем столбцам и строкам знакомые методы и операции.

DataFrame предоставляет широкие возможности для фильтрации данных по условиям, доступа к строкам и столбцам по меткам или позициям, изменения структуры таблицы, обработки пропущенных значений, вычисления агрегатных показателей и выполнения группировок. Умение эффективно работать с DataFrame является обязательным условием для дальнейшего изучения анализа данных, визуализации и машинного обучения.

Аппаратура и материалы:

Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленные библиотеки: `pandas`, `numpy`, `matplotlib`
Пример CSV-файл (предоставляется преподавателем или создаётся самостоятельно)
Дополнительно: встроенные датасеты `pandas` или `sklearn.datasets`

Типовые задачи:

1. Создайте DataFrame из словаря списков. Отобразите его размерность, названия столбцов и первых 5 строк.
2. Загрузите CSV-файл с помощью `pd.read_csv()` и выполните базовый анализ: типы данных, пропуски, статистика (`describe()`).
3. Выберите все строки, где значение в заданном числовом столбце превышает порог. Отсортируйте таблицу по этому столбцу по убыванию.

Контрольные вопросы:

1. Что такое DataFrame и чем он отличается от Series?
2. Как можно создать DataFrame вручную?
3. Как загрузить данные в DataFrame из внешнего файла?
4. Как получить доступ к отдельному столбцу и строке DataFrame?
5. Как узнать размерность, названия столбцов и типы данных в таблице?
6. Какие методы используются для фильтрации строк по условию?
7. Как сортируются данные в DataFrame?
8. Какие функции используются для анализа статистических характеристик данных?
9. Как проверяются и обрабатываются пропущенные значения?
10. Что такое индексация строк и зачем она может быть изменена?

ПРАКТИЧЕСКАЯ РАБОТА №6

Тема: Основы исследовательского анализа данных

Цель работы:

Научиться применять базовые методы исследовательского анализа данных (Exploratory Data Analysis, EDA) с использованием библиотеки pandas, визуализировать структуру данных и выявлять основные закономерности, пропуски и аномалии в датасете.

Теоретическое обоснование:

Исследовательский анализ данных (EDA) — это первый и важнейший этап в работе с данными, направленный на понимание структуры, свойств и особенностей набора данных. Цель EDA — выявить взаимосвязи между признаками, проверить гипотезы, определить уровень качества данных и сформировать представление о дальнейших этапах анализа или построения модели.

В рамках EDA используется множество инструментов: анализ типов данных и пропусков, вычисление описательных статистик, построение гистограмм, диаграмм рассеяния, boxplot-графиков, тепловых карт корреляций и др. Особое внимание уделяется визуализации, поскольку она позволяет наглядно представить скрытые зависимости, выбросы и ошибки в данных.

Инструментами исследовательского анализа являются функции библиотек `pandas`, `matplotlib`, `seaborn` и других. Проведение EDA повышает обоснованность выбора признаков, методов и моделей, используемых на последующих этапах машинного обучения.

Аппаратура и материалы:

Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленные библиотеки: `pandas`, `matplotlib`, `seaborn`, `numpy`
CSV-файл с реальными или синтетическими данными (например, `titanic.csv`, `iris.csv`)
Дополнительно: встроенные датасеты из `seaborn` (например, `tips`, `penguins`)

Типовые задачи:

1. Загрузите датасет (например, `titanic.csv`), определите его размерность, типы признаков, наличие пропусков, статистические характеристики.
2. Постройте визуализации: гистограммы для количественных переменных, boxplot для категориальных и scatterplot для пар признаков.
3. Постройте тепловую карту корреляции признаков и сделайте вывод о наличии линейных связей между ними.

Контрольные вопросы:

1. Что такое EDA и какова его роль в анализе данных?
2. Какие методы позволяют определить наличие пропущенных значений?
3. Какие статистические меры описывают распределение признаков?
4. Как визуализировать распределение количественной переменной?
5. Как интерпретировать диаграмму boxplot?
6. Что такое выбросы и как их можно обнаружить визуально?
7. Как исследовать зависимость между двумя количественными признаками?
8. Какие виды корреляции существуют и как их интерпретировать?
9. Зачем использовать тепловую карту и как она строится?
10. Как с помощью EDA можно выявить ошибки в структуре данных?

ПРАКТИЧЕСКАЯ РАБОТА №7

Тема: Линейная регрессия

Цель работы:

Освоить базовые принципы построения модели линейной регрессии с использованием библиотеки `scikit-learn`, научиться интерпретировать коэффициенты модели и оценивать её качество с помощью соответствующих метрик.

Теоретическое обоснование:

Линейная регрессия — один из простейших и наиболее распространённых методов машинного обучения. Она используется для моделирования зависимости между одной или

несколькими независимыми переменными (признаками) и зависимой переменной (целевой переменной), предполагая, что эта зависимость является линейной.

Модель линейной регрессии имеет вид:

$$y = w_0 + w_1x_1 + w_2x_2 + ... + w_nx_n,$$

где `w` — коэффициенты модели, определяемые в процессе обучения.

Основными этапами построения модели являются: разделение выборки на обучающую и тестовую части, обучение модели на обучающих данных, прогноз значений целевой переменной на тестовых данных и оценка точности прогноза. Основные метрики оценки качества модели — среднеквадратичная ошибка (MSE), средняя абсолютная ошибка (MAE) и коэффициент детерминации (R^2).

Линейная регрессия широко применяется в задачах прогноза, анализа трендов и интерпретации взаимосвязей между переменными. Важно уметь интерпретировать коэффициенты модели, оценивать её обобщающую способность и выявлять возможные проблемы, такие как мультиколлинеарность или переобучение.

Аппаратура и материалы:

Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленные библиотеки: `pandas`, `numpy`, `scikit-learn`, `matplotlib`, `seaborn`
Набор данных с числовыми признаками и непрерывной целевой переменной (например, `Boston`, `California housing`, `insurance.csv`)

Типовые задачи:

1. Загрузите датасет, выделите признаки и целевую переменную, разделите данные на обучающую и тестовую выборки.
2. Постройте модель линейной регрессии с использованием `LinearRegression`, выведите коэффициенты и интерпретируйте их.
3. Оцените качество модели с помощью метрик MSE, MAE и R^2, визуализируйте предсказания на тестовых данных.

Контрольные вопросы:

1. Что моделирует линейная регрессия и в чём её основное предположение?
2. Как интерпретируются коэффициенты линейной регрессии?
3. Какие этапы включает построение модели линейной регрессии?
4. Как осуществляется разделение данных на обучающую и тестовую выборки?
5. Что такое MSE, MAE и R^2, и как их интерпретировать?
6. Что показывает коэффициент детерминации R^2?
7. Как визуально оценить качество регрессионной модели?
8. Какие признаки могут указывать на переобучение модели?
9. В чём преимущества и ограничения линейной регрессии?
10. Как можно улучшить модель при наличии слабой линейной зависимости?

ПРАКТИЧЕСКАЯ РАБОТА №8

Тема: Логистическая регрессия

Цель работы:

Изучить принципы построения модели логистической регрессии для задач бинарной классификации, научиться обучать модель с использованием библиотеки `scikit-learn`, интерпретировать коэффициенты и оценивать её качество по метрике точности (accuracy).

Теоретическое обоснование:

Логистическая регрессия — базовый алгоритм классификации, применяемый для предсказания вероятности принадлежности объекта к одному из двух классов. Модель использует логистическую (сигмоидную) функцию, преобразующую линейную комбинацию входных признаков в значение от 0 до 1, которое интерпретируется как вероятность отнесения объекта к положительному классу.

Решающее правило модели основано на пороговом значении: если вероятность превышает заданный порог (по умолчанию — 0.5), объект классифицируется как положительный, иначе — как отрицательный. Коэффициенты модели отражают влияние соответствующих признаков на логарифм отношения шансов (log-odds).

В процессе обучения модель подбирает коэффициенты, максимизируя логарифмическую функцию правдоподобия. После обучения модель может быть использована для предсказания классов новых наблюдений. Основным критерием качества в данной работе является точность (accuracy), отражающая долю правильно классифицированных объектов.

Логистическая регрессия широко применяется в различных прикладных задачах, включая медицинскую диагностику, кредитный скоринг и маркетинг, благодаря своей интерпретируемости и эффективности на малых и средних по размеру выборках.

Аппаратура и материалы:

Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленные библиотеки: `pandas`, `numpy`, `scikit-learn`, `matplotlib`, `seaborn`
Набор данных для бинарной классификации (например, `breast_cancer` из `sklearn.datasets`, `titanic.csv`, `diabetes.csv`)

Типовые задачи:

1. Загрузите набор данных и выделите признаки и целевую переменную. Разделите данные на обучающую и тестовую выборки.
2. Постройте модель логистической регрессии с помощью `LogisticRegression`, обучите её и выведите коэффициенты.
3. Получите предсказания на тестовой выборке и рассчитайте точность модели с помощью метрики `accuracy_score`.

Контрольные вопросы:

1. В чём заключается основная идея логистической регрессии?
2. Как работает сигмоидная функция и почему она используется?
3. Какие данные подходят для логистической регрессии?
4. Как интерпретируются коэффициенты модели логистической регрессии?
5. Что означает порог классификации 0.5 и как его можно изменить?

6. Как формируется предсказание логистической модели?
7. Что такое точность (accuracy) и как она вычисляется?
8. Почему важно разделять данные на обучающую и тестовую выборки?
9. Как влияет масштабирование признаков на работу логистической регрессии?
10. Какие ограничения есть у логистической регрессии при работе с реальными данными?

ПРАКТИЧЕСКАЯ РАБОТА №9

Тема: Метрики качества моделей классификации

Цель работы:
Познакомиться с основными метриками оценки качества моделей классификации, научиться рассчитывать и интерпретировать точность, полноту, точность предсказания, F1-меру и строить confusion matrix с использованием библиотеки `scikit-learn`.

Теоретическое обоснование:
Оценка качества классификационных моделей необходима для понимания, насколько хорошо модель справляется с задачей предсказания классов. Использование только одной метрики, например точности (accuracy), может быть недостаточно, особенно в задачах с несбалансированными классами, где большинство объектов принадлежит одному классу. Для комплексной оценки качества классификации используются следующие метрики:

- **Точность (accuracy)** — доля правильных предсказаний от общего числа наблюдений.
- **Полнота (recall)** — доля объектов положительного класса, корректно предсказанных моделью.
- **Точность предсказания (precision)** — доля корректных положительных предсказаний среди всех предсказаний положительного класса.
- **F1-мера** — гармоническое среднее между precision и recall, применяется при необходимости сбалансировать обе метрики.
- **Матрица ошибок (confusion matrix)** — таблица, показывающая соотношение истинных и ложных положительных и отрицательных классификаций.

Понимание и правильная интерпретация этих метрик позволяет объективно сравнивать модели и делать обоснованный выбор подходящего алгоритма.

Аппаратура и материалы:
Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленные библиотеки: `pandas`, `numpy`, `scikit-learn`, `matplotlib`, `seaborn`
Набор данных для бинарной классификации (например, `breast_cancer`, `titanic.csv`, `diabetes.csv`)

Типовые задачи:
1. Обучите модель логистической регрессии или дерева решений на бинарном наборе данных и получите предсказания.
2. Постройте confusion matrix и визуализируйте её с помощью `seaborn.heatmap()`.
3. Рассчитайте и проанализируйте значения accuracy, precision, recall и F1-меры с помощью `classification_report()`.

Контрольные вопросы:
1. Что такое confusion matrix и какие значения она показывает?

2. Как рассчитывается точность (accuracy)?
3. В чём разница между precision и recall?
4. Что измеряет F1-мера и когда она особенно полезна?
5. Почему точность может быть недостаточной метрикой при несбалансированных классах?
6. Как интерпретировать высокое значение precision и низкое recall?
7. Что показывает `classification_report()` из `sklearn.metrics`?
8. Как визуализировать confusion matrix?
9. Какая метрика важнее — precision или recall — и от чего это зависит?
10. Как сравнить две модели по нескольким метрикам одновременно?

ПРАКТИЧЕСКАЯ РАБОТА №10

Тема: Метод ближайших соседей и метод опорных векторов

Цель работы:
Изучить принципы работы алгоритма k-ближайших соседей (kNN) и метода опорных векторов (SVM), научиться применять их для задач классификации с использованием библиотеки `scikit-learn`, настраивать параметры моделей и оценивать их точность.

Теоретическое обоснование:
Метод k-ближайших соседей (k-Nearest Neighbors) относится к числу ленивых алгоритмов обучения: модель не строится заранее, а предсказание делается на основе расстояния до ближайших объектов обучающей выборки. Алгоритм легко реализуем, не требует обучения, но чувствителен к масштабированию признаков и выбору числа соседей.

Метод опорных векторов (Support Vector Machine, SVM) относится к более сложным алгоритмам. Он ищет гиперплоскость, максимально разделяющую классы с учётом максимально возможного зазора (margin). При помощи ядерных функций метод может решать как линейные, так и нелинейные задачи классификации. SVM чувствителен к выбору параметров, особенно `C` и типа ядра (`kernel`), и требует предварительного масштабирования данных.

Оба метода используются в задачах классификации и распознавания, хорошо работают на задачах со средним числом признаков и наблюдений, но требуют аккуратной настройки и оценки.

Аппаратура и материалы:
Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленные библиотеки: `scikit-learn`, `pandas`, `numpy`, `matplotlib`, `seaborn`
Датасеты: `iris`, `digits`, `wine` из `sklearn.datasets` или другой набор для классификации

Типовые задачи:
1. Загрузите выбранный датасет, выполните масштабирование признаков с помощью `StandardScaler`.
2. Постройте модель `KNeighborsClassifier`, обучите её, подберите значение `k` (количество соседей), рассчитайте точность.
3. Постройте модель `SVC` с различными ядрами (`linear`, `rbf`), настройте параметр `C` и сравните точность моделей.

Контрольные вопросы:
1. Как работает метод k-ближайших соседей?

2. Что означает параметр `k` и как он влияет на поведение модели?
3. Почему масштабирование данных важно для метода kNN и SVM?
4. Как работает метод опорных векторов и что такое разделяющая гиперплоскость?
5. Что такое ядро (kernel) в SVM и какие типы ядер вы знаете?
6. Как параметр `C` влияет на обобщающую способность модели SVM?
7. Когда предпочтительнее использовать kNN, а когда — SVM?
8. Как оценить качество моделей на тестовой выборке?
9. Какие особенности есть у kNN с точки зрения производительности?
10. Как интерпретировать границы классов, полученные от модели SVM?

ПРАКТИЧЕСКАЯ РАБОТА №11

Тема: Деревья решений

Цель работы:
Освоить базовые принципы построения и анализа моделей классификации и регрессии на основе дерева решений, научиться визуализировать дерево, интерпретировать структуру модели и оценивать её точность.

Теоретическое обоснование:
Деревья решений являются интерпретируемыми и наглядными моделями, применяемыми как для задач классификации, так и регрессии. Основная идея метода заключается в последовательном разбиении пространства признаков на области, в которых принимается предсказание на основе преобладающего класса или среднего значения.

В процессе построения дерево выбирает признаки и пороговые значения, которые обеспечивают наилучшее разделение выборки по целевому признаку. Основными критериями качества разбиений являются индекс Джини, энтропия и дисперсия. Глубина дерева, минимальное количество объектов в узле и другие параметры регулируют сложность модели и позволяют избегать переобучения.

Модель дерева легко визуализируется и интерпретируется: каждый путь от корня к листу можно трактовать как набор логических условий, определяющих решение. Однако деревья склонны к переобучению, особенно на малых выборках, и чувствительны к шумам.

Аппаратура и материалы:
Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленные библиотеки: `scikit-learn`, `pandas`, `numpy`, `matplotlib`, `seaborn`, `graphviz` (опционально)
Датасеты: `iris`, `titanic`, `wine` или любой другой набор с категориальными и числовыми признаками

Типовые задачи:
1. Постройте модель классификации с использованием `DecisionTreeClassifier`, обучите её и выведите точность на тестовой выборке.
2. Визуализируйте структуру дерева с помощью `plot_tree()` или `export_graphviz()`, проанализируйте используемые признаки и уровни дерева.
3. Изучите влияние ограничения глубины дерева (`max_depth`) на качество модели.

Контрольные вопросы:
1. Как работает алгоритм построения дерева решений?
2. Какие критерии используются для оценки разбиения данных в узлах дерева?

3. Что такое переобучение и как оно проявляется в деревьях решений?
4. Как интерпретировать структуру дерева?
5. Что означает глубина дерева и как она влияет на модель?
6. Какие параметры позволяют управлять сложностью дерева?
7. Почему деревья решений чувствительны к незначительным изменениям в данных?
8. Чем отличается `DecisionTreeClassifier` от `DecisionTreeRegressor`?
9. Как визуализировать дерево и зачем это нужно?
10. Какие преимущества и ограничения имеют деревья решений по сравнению с другими моделями?

ПРАКТИЧЕСКАЯ РАБОТА №12

Тема: Случайный лес

Цель работы:
Изучить принцип работы ансамблевого метода «случайный лес», научиться строить модели классификации и регрессии с использованием `RandomForestClassifier` и `RandomForestRegressor`, интерпретировать важность признаков и оценивать точность модели.

Теоретическое обоснование:
Случайный лес (Random Forest) — это ансамблевый алгоритм, основанный на построении множества деревьев решений, каждое из которых обучается на случайной подвыборке исходных данных и с использованием случайного подмножества признаков. Итоговое предсказание формируется на основе голосования (для классификации) или усреднения (для регрессии) по всем деревьям.

Алгоритм сочетает простоту и интерпретируемость деревьев решений с высокой устойчивостью и способностью к обобщению, снижая вероятность переобучения. Он хорошо работает на задачах с большим числом признаков и наблюдений, не требует масштабирования данных и может быть использован как для классификации, так и для регрессии.

Одним из полезных свойств случайного леса является возможность оценки значимости признаков на основе их участия в разбиениях внутри деревьев. Это позволяет проводить отбор признаков и упрощать модели без значительной потери точности.

Аппаратура и материалы:
Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленные библиотеки: `scikit-learn`, `pandas`, `numpy`, `matplotlib`, `seaborn`
Датасеты: `breast_cancer`, `wine`, `titanic`, `diabetes`, или любой CSV-файл с метками классов

Типовые задачи:
1. Обучите модель `RandomForestClassifier` на выбранном датасете, рассчитайте точность на тестовой выборке.
2. Постройте график важности признаков с помощью `feature_importances_` и визуализируйте его.
3. Изучите влияние числа деревьев (`n_estimators`) и глубины (`max_depth`) на качество модели.

Контрольные вопросы:
1. В чём заключается идея метода случайного леса?

2. Чем случайный лес отличается от одного дерева решений?
3. Как происходит обучение отдельных деревьев в случайном лесе?
4. Что означает параметр `n_estimators`?
5. Какие признаки отбираются при обучении каждого дерева?
6. Как рассчитывается итоговое предсказание в случайном лесе?
7. Как определяется важность признаков в модели?
8. Какие параметры случайного леса влияют на переобучение?
9. Какие преимущества имеет случайный лес перед другими моделями классификации?
10. В каких задачах случайный лес особенно эффективен?

ПРАКТИЧЕСКАЯ РАБОТА №13

Тема: Градиентный бустинг

Цель работы:
Изучить принципы работы метода градиентного бустинга, научиться обучать модель с помощью `GradientBoostingClassifier` из библиотеки `scikit-learn`, оценивать точность и анализировать влияние параметров модели на её качество.

Теоретическое обоснование:
Градиентный бустинг — это ансамблевый метод, который строит последовательность слабых моделей (обычно деревьев решений), каждая из которых обучается на ошибках предыдущей. Каждое последующее дерево корректирует предсказания модели, стремясь минимизировать заданную функцию потерь.

Отличительной особенностью градиентного бустинга является его высокая точность и способность выявлять сложные нелинейные зависимости. Алгоритм чувствителен к выбору параметров, таких как количество деревьев (`n_estimators`), скорость обучения (`learning_rate`) и глубина деревьев (`max_depth`). При правильной настройке он может превосходить по качеству более простые модели, но требует регулярной валидации для предотвращения переобучения.

Градиентный бустинг используется в широком спектре задач, включая кредитный скоринг, анализ оттока клиентов, медицинскую диагностику и соревнования по машинному обучению.

Аппаратура и материалы:
Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленные библиотеки: `scikit-learn`, `pandas`, `numpy`, `matplotlib`, `seaborn`
Набор данных: `breast_cancer`, `titanic`, `wine`, или собственный CSV-файл для классификации

Типовые задачи:
1. Постройте модель `GradientBoostingClassifier` на выбранном наборе данных, обучите её и рассчитайте точность на тестовой выборке.
2. Измените значения параметров `n_estimators`, `learning_rate` и `max_depth`, проанализируйте, как они влияют на точность модели.
3. Постройте график важности признаков и сделайте вывод о значимых переменных.

Контрольные вопросы:
1. Как работает метод градиентного бустинга?
2. Чем градиентный бустинг отличается от случайного леса?

3. Что означает параметр `learning_rate` и как он влияет на модель?
4. Как параметр `n_estimators` влияет на переобучение и недообучение?
5. В чём роль деревьев в градиентном бустинге?
6. Какие функции потерь могут использоваться в градиентном бустинге?
7. Какова роль регуляризации в бустинге?
8. Что показывает `feature_importances_` в модели бустинга?
9. Как определить оптимальные параметры модели?
10. В каких задачах градиентный бустинг может быть предпочтительнее других методов?

ПРАКТИЧЕСКАЯ РАБОТА №14

Тема: Обучение без учителя: задачи кластеризации данных

Цель работы:

Познакомиться с методами кластеризации данных, освоить работу с алгоритмами `KMeans` и `AgglomerativeClustering` из библиотеки `scikit-learn`, научиться визуализировать результаты кластеризации и интерпретировать структуру полученных кластеров.

Теоретическое обоснование:

Обучение без учителя применяется в ситуациях, когда отсутствуют заранее размеченные классы. Кластеризация — один из ключевых подходов в этом типе обучения — заключается в разделении объектов на группы (кластеры) по схожести признаков без предварительного знания истинных меток классов.

Метод **k-средних (KMeans)** разбивает данные на заданное количество кластеров, минимизируя внутрикластерное расстояние. Этот метод прост, интерпретируем и хорошо масштабируется, но требует заранее определить число кластеров.

Метод **иерархической кластеризации (AgglomerativeClustering)** строит древовидную структуру объединения кластеров, что позволяет исследовать данные на разных уровнях иерархии. Кластеры формируются путём последовательного объединения ближайших объектов или групп объектов.

Кластеризация применяется в сегментации пользователей, поиске аномалий, биоинформатике, анализе текстов и других задачах, где важно выявить внутреннюю структуру данных.

Аппаратура и материалы:

Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленные библиотеки: `scikit-learn`, `pandas`, `numpy`, `matplotlib`, `seaborn`
Датасеты: `iris`, `wine`, синтетические выборки из `sklearn.datasets.make_blobs` или `make_moons`

Типовые задачи:

1. Выполните кластеризацию с помощью `KMeans` на выбранном датасете, визуализируйте кластеры и сравните их с истинными метками (если они известны).
2. Постройте иерархическую кластеризацию с помощью `AgglomerativeClustering`, отобразите дендрограмму и интерпретируйте число кластеров.
3. Проведите уменьшение размерности с использованием PCA для визуализации результатов кластеризации в 2D.

Контрольные вопросы:

1. Чем обучение без учителя отличается от обучения с учителем?
2. В чём состоит основная идея метода k-средних?
3. Как выбрать оптимальное число кластеров в KMeans?
4. Как работает иерархическая кластеризация?
5. Что такое дендрограмма и как её интерпретировать?
6. Какие меры используются для оценки качества кластеризации?
7. Почему результаты кластеризации чувствительны к масштабу признаков?
8. Что такое внутрикластерное расстояние и как оно используется в KMeans?
9. Как влияет начальная инициализация центров кластеров на итоговый результат?
10. Какие реальные задачи можно решать методами кластеризации?

ПРАКТИЧЕСКАЯ РАБОТА №15

Тема: Обучение без учителя: задачи снижения размерности данных

Цель работы:

Изучить методы снижения размерности данных, научиться применять алгоритмы `PCA` (анализ главных компонент) и `t-SNE` для упрощения структуры данных и визуализации многомерных признаковых пространств.

Теоретическое обоснование:

Задачи снижения размерности направлены на представление данных с большим числом признаков в пространстве меньшей размерности при сохранении как можно большей части информации. Это позволяет упростить последующий анализ, визуализировать данные, уменьшить шум и вычислительную сложность, а также подготовить признаки для машинного обучения.

Метод **PCA (Principal Component Analysis)** — линейный алгоритм, основанный на преобразовании признаков в новую ортогональную базу, упорядоченную по убыванию дисперсии. PCA используется для компрессии данных, устранения мультиколлинеарности и визуализации в 2D/3D.

Метод **t-SNE (t-distributed Stochastic Neighbor Embedding)** — нелинейный алгоритм, ориентированный на сохранение локальной структуры данных. Он часто используется для визуализации кластеров и структуры сложных многомерных пространств, особенно при работе с высокоразмерными и плохо интерпретируемыми данными.

Методы снижения размерности играют важную роль в предварительном анализе данных, помогают выявлять скрытые зависимости и облегчать интерпретацию моделей.

Аппаратура и материалы:

Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленные библиотеки: `scikit-learn`, `pandas`, `numpy`, `matplotlib`, `seaborn`
Датасеты: `iris`, `digits`, `wine`, `make_blobs` или собственные CSV-файлы

Типовые задачи:

1. Выполните PCA на выбранном датасете, определите количество компонент, сохраняющих ≥90% дисперсии, и визуализируйте данные в 2D-пространстве.
2. Выполните t-SNE на том же датасете, сравните распределение точек на диаграмме с результатом PCA.
3. Постройте график объяснённой дисперсии для первых компонент PCA и прокомментируйте результат.

Контрольные вопросы:

1. В чём заключается цель снижения размерности?
2. Как работает алгоритм PCA?
3. Что такое главные компоненты и как они интерпретируются?
4. Как определить, сколько компонент нужно оставить при использовании PCA?
5. В чём отличие линейного метода PCA от нелинейного t-SNE?
6. Какие параметры t-SNE влияют на результат визуализации?
7. Как визуализация после PCA может помочь в понимании структуры данных?
8. В каких задачах целесообразно использовать снижение размерности?
9. Какие ограничения есть у метода PCA?
10. Зачем применять методы снижения размерности перед кластеризацией или классификацией?

ПРАКТИЧЕСКАЯ РАБОТА №16

Тема: Подбор гиперпараметров моделей

Цель работы:

Научиться настраивать гиперпараметры моделей машинного обучения с использованием методов перебора по сетке (`GridSearchCV`) и случайного поиска (`RandomizedSearchCV`), освоить применение кросс-валидации для повышения надёжности оценки качества модели.

Теоретическое обоснование:

Гиперпараметры — это параметры модели, значение которых не определяется в процессе обучения, а задаётся вручную до начала обучения. К ним относятся, например, количество соседей в `KNeighborsClassifier`, глубина дерева решений в `DecisionTreeClassifier`, параметр регуляризации `C` в логистической регрессии и др.
Правильный выбор гиперпараметров существенно влияет на обобщающую способность модели. Один из подходов к их подбору — использование **кросс-валидации**, при которой обучающая выборка делится на несколько частей, и модель оценивается на каждой из них.
Для автоматизированного подбора параметров используются:

- `GridSearchCV` — полный перебор всех возможных комбинаций заданных значений параметров;
- `RandomizedSearchCV` — случайный перебор заданного числа случайных комбинаций.

Обе процедуры позволяют найти наилучшую конфигурацию модели на основе заданной метрики качества, обычно — точности (accuracy). Такой подход повышает устойчивость модели и помогает избежать переобучения.

Аппаратура и материалы:

Компьютер или ноутбук с доступом в интернет
Jupyter Notebook или Google Colab
Установленные библиотеки: `scikit-learn`, `pandas`, `numpy`, `matplotlib`, `seaborn`
Датасеты: `iris`, `wine`, `titanic`, или любой CSV-файл для классификации

Типовые задачи:

1. Настройте параметры модели `KNeighborsClassifier` с использованием `GridSearchCV`, определите оптимальное значение `n_neighbors`.

2. Примените `RandomizedSearchCV` к модели дерева решений, подбирая `max_depth` и `min_samples_split`, и проанализируйте результат.
3. Сравните точность модели с оптимальными параметрами и модели с параметрами по умолчанию.

Контрольные вопросы:

1. Что такое гиперпараметры и чем они отличаются от параметров модели?
2. Зачем выполнять подбор гиперпараметров?
3. В чём разница между `GridSearchCV` и `RandomizedSearchCV`?
4. Что такое кросс-валидация и как она работает?
5. Как выбрать диапазоны значений для перебора?
6. Какие метрики можно использовать при подборе гиперпараметров?
7. Как использовать полученную модель после подбора параметров?
8. Какие риски связаны с избыточным подбором гиперпараметров?
9. Что такое переобучение при подборе параметров и как его избежать?
10. Какие способы ускорения подбора гиперпараметров существуют?

ЗАКЛЮЧЕНИЕ

Практические работы, представленные в данных методических указаниях, охватывают ключевые направления дисциплины *«Искусственный интеллект и машинное обучение»* и направлены на формирование у студентов устойчивых навыков построения, настройки и анализа моделей машинного обучения с использованием современных инструментов и библиотек языка Python.

Структура работ выстроена по нарастающей сложности: от освоения базовых инструментов программной среды (Jupyter Notebook, NumPy, pandas, matplotlib) к практическому применению алгоритмов обучения с учителем и без учителя, включая методы классификации, регрессии, ансамблевого обучения, кластеризации и снижения размерности. Завершающие занятия посвящены вопросам повышения эффективности моделей — выбору метрик качества и подбору гиперпараметров.

Каждая практическая работа содержит цель, теоретическое обоснование, список необходимого программного обеспечения и данных, типовые задачи для выполнения, а также контрольные вопросы для самопроверки и текущего контроля.

Систематическое выполнение практических заданий способствует: – закреплению теоретических знаний; – формированию прикладных компетенций по анализу и интерпретации данных; – развитию аналитического мышления и умения работать с реальными задачами; – подготовке к самостоятельной профессиональной и исследовательской деятельности в области анализа данных и искусственного интеллекта.

Практическая часть курса ориентирована на использование открытых и доступных программных инструментов, что позволяет обеспечить гибкость обучения и применимость полученных знаний в самых разных сферах.

УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература:

Болотский, А. В.; Исследование операций и методы оптимизации Электронный ресурс / Болотский А. В., Кочеткова О. А. - Санкт-Петербург : Лань, 2020. - 116 с. - ISBN 978-5-8114-4568-4

Сараев, П. В.; Методы машинного обучения Электронный ресурс : Методические указания и задания к лабораторным работам по курсу / П. В. Сараев. - Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2017. - 48 с. - Книга находится в

премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397

Сопов, Е. А.; Многокритериальные нейроэволюционные системы в задачах машинного обучения и человеко-машинного взаимодействия Электронный ресурс / Сопов Е. А., Иванов И. А. : монография. - Красноярск : СФУ, 2019. - 160 с. - ISBN 978-5-7638-3969-2

Дополнительная литература:

Методические рекомендации к практическим занятиям по дисциплине "Теория и практика машинного перевода" : Направление подготовки 45.04.02 Лингвистика. Магистерская программа "Перевод и переводоведение". Квалификация выпускника - магистр. Форма обучения - очная. Учебный план 2015 года. Изучается в 3 семестре. - Ставрополь : СКФУ, 2015. - 9 с.

Неделько, В. М.; Основы статистических методов машинного обучения Электронный ресурс : Учебное пособие / В. М. Неделько. - Новосибирск : Новосибирский государственный технический университет, 2010. - 72 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-7782-1385-2

Рапаков, Г. Г.; Методы и алгоритмы машинного обучения при принятии управленческих решений в региональной системе медицинской профилактики (опыт Вологодской области) Электронный ресурс / Рапаков Г. Г., Касимов Р. А. : монография. - Вологда : ВоГУ, 2014. - 143 с. - ISBN 978-5-87851-548-1

Учебно-методическая литература:

Сараев, П. В.; Методы машинного обучения Электронный ресурс : Методические указания и задания к лабораторным работам по курсу / П. В. Сараев. - Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2017. - 48 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по организации и проведению самостоятельной работы
по дисциплине «Искусственный интеллект и машинное обучение»
для студентов направления подготовки
09.03.01 Информатика и вычислительная техника
Направленность (профиль)
«Программное обеспечение средств вычислительной техники и автоматизированных систем»

Ставрополь 2026

Содержание

ВВЕДЕНИЕ

1. Цель и задачи освоения дисциплины (модуля)

Цель дисциплины – формирование, согласно образовательной программы, компетенций ПК-1, ПК-2 обучающегося.

Для достижения этой цели в процессе изучения дисциплины решаются следующие задачи: ознакомление с основными понятиями и типами задач машинного обучения; изучение методов подготовки данных, включая очистку, нормализацию и инженерное преобразование признаков; овладение базовыми моделями обучения с учителем, такими как линейная и логистическая регрессия; освоение стохастического градиентного спуска как метода оптимизации моделей; формирование представлений о метриках оценки качества классификации и интерпретации результатов моделей; изучение методов борьбы с переобучением, включая регуляризацию и кросс-валидацию; освоение алгоритмов классификации на основе метода ближайших соседей и метода опорных векторов; изучение алгоритмов построения деревьев решений и ансамблевых моделей, таких как случайный лес и градиентный бустинг; знакомство с методами обучения без учителя, включая алгоритмы кластеризации; овладение методами уменьшения размерности и визуализации данных, такими как PCA и t-SNE; развитие практических навыков настройки моделей с использованием подбора гиперпараметров; формирование умений сравнивать и интерпретировать результаты различных моделей на основе объективных метрик.

2. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина относится к части, формируемой участниками образовательных отношений.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ПК-1. Способен разрабатывать требования и проектировать программное обеспечение	ИД-2$_{ПК-1}$ проводит анализ исполнения требований; вырабатывает варианты реализации требований; проводит оценку и обоснование рекомендуемых решений; выбирает средства реализации требований к программному обеспечению; использует существующие типовые решения и шаблоны проектирования программного обеспечения; разрабатывает технические спецификации на программные компоненты и их взаимодействие.	Проведение анализа исполнения требований; выработка вариантов реализации требований; проведение оценки и обоснование рекомендуемых решений; выбор средств реализации требований к программному обеспечению; использование существующие типовых решений и шаблонов проектирования программного обеспечения; разработка технических

		спецификаций на программные компоненты и их взаимодействие.
ПК-2. Способен осуществлять концептуальное, функциональное и логическое проектирование систем среднего и крупного масштаба и сложности	ИД-1$_{ПК-2}$ реализует методы планирования проектных работ; применяет методы классического системного анализа; использует теорию управления бизнес-процессами, методы концептуального проектирования, методы оценки качества программных систем; осуществляет решение задач оптимизации при разработке бизнес-требований к системе.	Понимание методов классического системного анализа; понимание теории управления бизнес-процессами; понимание методов концептуального проектирования; понимание методов оценки качества программных систем.
	ИД-2$_{ПК-2}$ разрабатывает технико-экономическое обоснование; разрабатывает техническое задание на систему; разрабатывает требования к подсистемам системы и осуществляет контроль их качества; организовывает оценку соответствия требованиям существующих систем и их аналогов; выполняет сопровождение приемочных испытаний и ввод в эксплуатацию системы; обрабатывает запросы на изменение требований к системе.	Способность: разработать технико-экономическое обоснование; разработать техническое задание на систему; разработать требования к подсистемам системы и осуществить контроль их качества; организовать оценку соответствия требованиям существующих систем и их аналогов; выполнить сопровождение приемочных испытаний и ввод в эксплуатацию системы; обработать запросы на изменение требований к системе.
	ИД-3$_{ПК-2}$ составляет графики контрольных мероприятий; разрабатывает бизнес-требования к системе; способен определять ключевые свойства и ограничения системы; выделяет подсистемы системы	Способность составлять графики контрольных мероприятий; разрабатывать бизнес-требования к системе; способен определять ключевые свойства и ограничения системы; выделяет подсистемы системы
	ИД-4$_{ПК-2}$ Осуществляет решение задач оптимизации при разработке бизнес-требований к системе	Способность решать оптимизационные задачи при проектировании систем машинного обучения.

	ИД-12$_{ПК-2}$ Разрабатывает программный код на языках программирования высокого уровня	Способен разрабатывать системы машинного обучения на современных языках высокого уровня.
	ИД-15$_{ПК-2}$ Осуществляет сбор, обработку и анализ результатов оценки готовых систем на соответствие требований	Осуществление сбора обработки и анализа результатов готовых систем машинного обучения на соответствие требованиям.

4. Объем учебной дисциплины (модуля) и формы контроля *

Объем занятий: всего: 3 з.е. 108 акад.ч.	ОФО, В акад. часах
Контактная работа:	64.00/32.00
Лекции/из них практическая подготовка	32.00/0
Лабораторных работ/из них практическая подготовка	32.00/32.00
Практических занятий/из них практическая подготовка	
Самостоятельная работа	44.00
Формы контроля	
Зачет с оценкой	да

САМОСТОЯТЕЛЬНАЯ РАБОТА КАК ВАЖНЕЙШАЯ ФОРМА УЧЕБНОГО ПРОЦЕССА

Самостоятельная работа студентов является неотъемлемой частью освоения дисциплины *«Искусственный интеллект и машинное обучение»* и играет ключевую роль в формировании устойчивых и глубоких знаний. Современные методы машинного обучения требуют не только теоретического понимания алгоритмов, но и уверенного владения инструментами, навыков анализа данных, интерпретации результатов и критического мышления.

В процессе самостоятельной работы студент получает возможность:

- закрепить и углубить материал, изученный на лекциях и практических занятиях;
- научиться самостоятельно находить и изучать профессиональные источники информации;
- применять изученные алгоритмы к реальным или приближённым к реальности задачам;
- развивать практические навыки программирования, работы с библиотеками Python и визуализации данных;
- осваивать различные подходы к решению одной и той же задачи, выбирать оптимальные методы, обосновывать сделанный выбор;
- готовиться к итоговому контролю, демонстрируя уверенность в базовых понятиях и алгоритмах.

Кроме того, самостоятельная работа развивает дисциплину, инициативность, умение планировать собственное обучение и адаптироваться к постоянно развивающейся предметной области. Эти навыки особенно важны в сфере искусственного интеллекта, где темпы технологических изменений требуют постоянного самообновления знаний.

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного

участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения. Обучение в ВУЗе включает в себя две, практически одинаковые по объему и взаимовлиянию части – процесса обучения и процесса самообучения. Поэтому СРС должна стать эффективной и целенаправленной работой студента.

Концепцией модернизации российского образования определены основные задачи профессионального образования - "подготовка квалифицированного работника соответствующего уровня и профиля, конкурентоспособного на рынке труда, компетентного, ответственного, свободно владеющего своей профессией и ориентированного в смежных областях деятельности, способного к эффективной работе по специальности на уровне мировых стандартов, готового к постоянному профессиональному росту, социальной и профессиональной мобильности".

Решение этих задач невозможно без повышения роли самостоятельной работы студентов над учебным материалом, усиления ответственности преподавателей за развитие навыков самостоятельной работы, за стимулирование профессионального роста студентов, воспитание творческой активности и инициативы.

К современному специалисту общество предъявляет достаточно широкий перечень требований, среди которых немаловажное значение имеет наличие у выпускников определенных способностей и умения самостоятельно добывать знания из различных источников, систематизировать полученную информацию, давать оценку конкретной финансовой ситуации. Формирование такого умения происходит в течение всего периода обучения через участие студентов в лабораторных занятиях, выполнение контрольных заданий и тестов, написание курсовых и выпускных квалификационных работ. При этом самостоятельная работа студентов играет решающую роль в ходе всего учебного процесса.

Формы самостоятельной работы студентов разнообразны. По данной дисциплине «Искусственный интеллект и машинное обучение» предусмотрены следующие:

Использование материала учебно-методического комплекса дисциплины

На первом этапе необходимо ознакомиться с обязательным минимумом содержания основной образовательной программы по изучаемой дисциплине. Далее необходимо изучить рабочую программу дисциплины, в которой рассмотрено содержание тем дисциплины лекционного курса, взаимосвязь тем лекций с лабораторных занятий, темы и виды самостоятельной работы. По каждому виду самостоятельной работы предусмотрены определённые формы отчетности, представленные в учебной программе по дисциплине «Искусственный интеллект и машинное обучение».

Работа с литературой

Для успешного освоения дисциплины, необходимо самостоятельно детально изучить представленные темы по рекомендуемым источникам информации, которые представлены в учебной программе.

Самостоятельная работа приобщает студентов к научному творчеству, поиску и решению актуальных современных проблем.

ЦЕЛИ И ОСНОВНЫЕ ЗАДАЧИ СРС

Цели самостоятельной работы:

- Углубление теоретических знаний, полученных на лекциях;
- Закрепление практических навыков, полученных на семинарах;
- Развитие умений решать прикладные задачи машинного обучения на практике;
- Формирование критического подхода к выбору моделей, параметров и метрик.

Формы самостоятельной работы:

- Изучение рекомендованных источников и материалов лекций;
- Анализ и решение предложенных задач;
- Выполнение индивидуальных практических заданий в Jupyter Notebook;
- Подготовка мини-отчётов и интерпретация полученных результатов;
- Самостоятельное повторение материала и подготовка к тестированию.

Ведущая цель организации и осуществления СРС должна совпадать с целью обучения студента – подготовкой бакалавра с высшим образованием. При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности.

Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

ВИДЫ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

В образовательном процессе высшего профессионального образовательного учреждения выделяется два вида самостоятельной работы – аудиторная, под руководством преподавателя, и внеаудиторная. Тесная взаимосвязь этих видов работ предусматривает дифференциацию и эффективность результатов ее выполнения и зависит от организации, содержания, логики учебного процесса (межпредметных связей, перспективных знаний и др.):

Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию.

Внеаудиторная самостоятельная работа выполняется студентом по заданию преподавателя, но без его непосредственного участия.

Основными видами самостоятельной работы студентов без участия преподавателя по данной дисциплине являются:

- формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);
- подготовка к лабораторным работам, их оформление.

Основными видами самостоятельной работы студентов с участием преподавателей являются:

- текущие консультации;
- прием и защита лабораторных работ (во время проведения л/р).

ОРГАНИЗАЦИЯ СРС

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей дисциплины «Искусственный интеллект и машинное обучение», объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Рекомендуемые темы и задания для самостоятельной проработки:

1. Подготовка данных и визуализация

- Изучить типы признаков, методы масштабирования и кодирования.
- **Задание**: выбрать набор данных из `sklearn.datasets` (например, `wine` или `breast_cancer`), визуализировать распределение признаков (гистограммы, парные графики), применить `StandardScaler` и `OneHotEncoder`.

2. Анализ линейной и логистической регрессии

- Изучить математические основы и методы оценки качества.
- **Задание**: обучить модель линейной регрессии на данных Boston housing, рассчитать MAE и R^2; построить логистическую регрессию на `breast_cancer` и построить ROC-кривую.

3. Переобучение и регуляризация

- Изучить понятия L1 и L2-регуляризации, значение параметра `C`.
- **Задание**: на датасете `digits` обучить логистическую регрессию с разными значениями регуляризации и сравнить результаты на обучающей и тестовой выборке.

4. Методы классификации: KNN и SVM

- Сравнить работу моделей с разными параметрами.
- **Задание:** реализовать KNN и SVM на одинаковом наборе данных (`iris`), построить графики точности в зависимости от параметров (`k`, `C`, `kernel`).

5. Ансамбли моделей

- Изучить основы случайного леса и градиентного бустинга.

- **Задание:** обучить модели `RandomForestClassifier` и `GradientBoostingClassifier`, сравнить их точность и важность признаков (`feature_importances_`).

6. Методы кластеризации

- Изучить отличие K-средних от иерархической кластеризации.
- **Задание**: применить KMeans и Agglomerative Clustering к `iris`, сравнить результаты с истинными метками. Построить дендрограмму.

7. Уменьшение размерности и визуализация

- Изучить алгоритмы PCA и t-SNE.
- **Задание:** уменьшить размерность `digits` с помощью PCA до 2 компонент и визуализировать результат. Повторить с t-SNE.

8. Подбор гиперпараметров

- Освоить работу с `GridSearchCV`.
- **Задание:** подобрать оптимальные параметры для SVM с помощью `GridSearchCV` на датасете `wine`. Зафиксировать метрику оценки.

Типовые задания для СРС:

Общий подход к оцениванию:

- Каждое задание оценивается по **10-балльной шкале**;
- Общая оценка по блоку самостоятельной работы может рассчитываться как среднее значение или сумма (по усмотрению преподавателя);
- Возможно использование **частичных баллов (0.5, 1 и т.д.)** при частично выполненных пунктах задания;
- При копировании без осмысления, неработающем коде или отсутствующей интерпретации баллы **не начисляются**.

№	Задание	Критерии оценки (макс. 10 баллов)
1	Изучить отличия между обучением с учителем и без учителя. Привести по 2 примера задач каждого типа из профессиональной области.	2 балла — корректное теоретическое определение; 4 балла — уместные примеры; 4 балла — пояснение и анализ различий.
2	Загрузить набор данных `iris` или `wine`. Выполнить предварительный анализ: размерность, типы признаков, распределения.	3 балла — загрузка и описание данных; 3 балла — визуализация распределений; 4 балла — интерпретация результатов.
3	Масштабировать признаки с помощью `StandardScaler` и `MinMaxScaler`. Сравнить влияние масштабирования.	3 балла — корректная реализация; 3 балла — визуализация изменений; 4 балла — аналитическое сравнение.
4	Закодировать категориальные признаки с помощью `OneHotEncoder` и `OrdinalEncoder`, пояснить разницу.	3 балла — корректная реализация; 3 балла — примеры; 4 балла — пояснение различий и когда что применять.
5	Построить модель линейной регрессии на Boston housing. Рассчитать MAE, MSE, R^2.	2 балла — подготовка данных; 4 балла — реализация модели и метрик; 4 балла — интерпретация результатов.

6	Построить логистическую регрессию на `breast_cancer`, рассчитать метрики, построить ROC-кривую.	3 балла — обучение модели; 4 балла — корректные расчёты метрик; 3 балла — ROC-кривая и выводы.
7	Реализовать градиентный спуск вручную для линейной регрессии, построить график функции ошибки.	4 балла — реализация без sklearn; 3 балла — визуализация; 3 балла — интерпретация графика.
8	Сравнить работу KNN и SVM на одном датасете, визуализировать и сравнить метрики.	3 балла — корректная реализация; 3 балла — визуализация; 4 балла — аналитическое сравнение.
9	Построить дерево решений на `iris`, визуализировать, проанализировать признаки переобучения.	3 балла — построение и визуализация; 3 балла — анализ структуры дерева; 4 балла — признаки переобучения.
10	Обучить `RandomForestClassifier`, сравнить со стандартным деревом.	3 балла — корректная реализация обеих моделей; 3 балла — сравнение точности; 4 балла — анализ важности признаков.
11	Обучить `GradientBoostingClassifier` на `wine`, подобрать параметры.	3 балла — обучение модели; 3 балла — перебор параметров; 4 балла — обоснование выбора и сравнение.
12	Провести кросс-валидацию для логистической регрессии, сравнить с train/test.	3 балла — реализация CV; 3 балла — сравнение результатов; 4 балла — выводы о стабильности модели.
13	Подобрать гиперпараметры SVM с помощью `GridSearchCV`, представить таблицу.	3 балла — реализация поиска; 3 балла — оформление таблицы; 4 балла — обоснование выбора.
14	Кластеризация K-средних на `iris`, сравнение с реальными метками, silhouette score.	3 балла — реализация; 3 балла — сравнение кластеров и меток; 4 балла — оценка качества.
15	Иерархическая кластеризация, дендрограмма, определение числа кластеров.	3 балла — построение модели; 3 балла — визуализация дендрограммы; 4 балла — корректная интерпретация.
16	Применить PCA к `digits`, визуализировать в 2D.	3 балла — реализация PCA; 3 балла — визуализация; 4 балла — интерпретация изменений размерности.
17	Сравнить визуализации PCA и t-SNE. Пояснить выявленные структуры.	3 балла — реализация и графики; 3 балла — сравнение результатов; 4 балла — объяснение различий.
18	Изучить влияние регуляризации на логистическую регрессию, построить график зависимости точности от C.	3 балла — эксперимент с C; 3 балла — график; 4 балла — интерпретация и выводы.
19	Составить таблицу по алгоритмам классификации: плюсы, минусы, применение.	3 балла — полнота таблицы; 3 балла — корректность сведений; 4 балла — чёткость формулировок и практичность.

20	Подготовить Jupyter Notebook с полным циклом МЛ на произвольном датасете.	2 балла — выбор и загрузка данных; 4 балла — выполнение всех этапов (предобработка, модель, метрики); 4 балла — логичность структуры и оформление.
21	Выбрать любой набор данных из `sklearn.datasets` и провести разведочный анализ данных (EDA).	4 — корректный и разнообразный EDA; 3 — выводы; 3 — аккуратное оформление и визуализация.
22	Реализовать код, проверяющий наличие пропущенных значений и выбросов.	3 — код проверки; 3 — визуализация выбросов; 4 — интерпретация.
23	Визуализировать корреляции признаков с помощью тепловой карты (`seaborn.heatmap`).	3 — корректный код; 3 — наглядная визуализация; 4 — выводы о мультиколлинеарности.
24	Сравнить методы масштабирования: `StandardScaler`, `MinMaxScaler`, `RobustScaler`.	4 — применение всех методов; 3 — визуализация; 3 — выводы по различиям.
25	Применить `PolynomialFeatures` и обучить полиномиальную регрессию.	3 — генерация новых признаков; 4 — обучение и визуализация; 3 — выводы.
26	Реализовать функцию собственного кросс-валидационного разбиения (без sklearn).	5 — реализация; 2 — корректность; 3 — пояснение логики.
27	Исследовать влияние несбалансированных классов на точность логистической регрессии.	4 — создание/загрузка несбалансированного набора; 3 — сравнение; 3 — выводы.
28	Использовать `SMOTE` для балансировки классов.	3 — корректное применение; 4 — сравнение результатов до и после; 3 — выводы.
29	Применить классификатор на случайно сгенерированных данных (`make_classification`).	3 — генерация набора; 4 — обучение и метрики; 3 — интерпретация сложности задачи.
30	Исследовать влияние количества признаков на производительность модели.	3 — создание версий датасета; 4 — обучение; 3 — график/анализ зависимости.
31	Реализовать жадный отбор признаков (`SelectKBest`) и сравнить с моделью без отбора.	3 — реализация; 4 — сравнение точности; 3 — выводы.
32	Применить `Pipeline` из `sklearn` для полного цикла: масштабирование + модель.	4 — корректная структура пайплайна; 3 — выводы; 3 — рефакторинг кода.
33	Исследовать влияние outliers на точность KNN.	3 — создание набора с выбросами; 4 — метрики; 3 — анализ поведения модели.
34	Провести классификацию при помощи `NaiveBayes`, сравнить с логистической регрессией.	3 — обучение моделей; 4 — метрики и таблица; 3 — выводы.
35	Реализовать метод голосования (`VotingClassifier`) на основе 3 моделей.	3 — сбор ансамбля; 4 — сравнение точности; 3 — объяснение эффекта.

36	Реализовать модель случайного леса с использованием только наиболее важных признаков.	3 — определение важности; 4 — отбор признаков; 3 — сравнение качества.
37	Провести регрессию с использованием `Ridge` и `Lasso`.	3 — реализация; 4 — графики весов; 3 — сравнение и интерпретация.
38	Построить график зависимости точности классификации от доли обучающей выборки.	4 — создание разбиений; 3 — построение графика; 3 — выводы.
39	Сравнить точность моделей с нормализованными и ненормализованными признаками.	3 — сравнение; 4 — таблица результатов; 3 — объяснение различий.
40	Написать собственную реализацию логистической регрессии (без sklearn).	4 — реализация; 3 — тестирование; 3 — сравнение с `sklearn`.
41	Провести визуализацию решающих границ для SVM и KNN на двумерных данных.	4 — график; 3 — аннотация; 3 — пояснение особенностей.
42	Реализовать сравнение моделей по времени обучения и предсказания.	3 — корректные замеры; 4 — график или таблица; 3 — интерпретация.
43	Изучить влияние шума (noise) в данных на классификацию.	3 — добавление шума; 4 — сравнение моделей; 3 — выводы.
44	Сравнить модели по устойчивости к переобучению на маленьком и большом объёме данных.	4 — реализация; 3 — визуализация результатов; 3 — анализ.
45	Реализовать обучающую и валидационную кривые (`learning_curve`) для SVM.	3 — код; 4 — графики; 3 — интерпретация.
46	Построить confusion matrix и проанализировать ошибки модели.	3 — реализация; 4 — визуализация; 3 — интерпретация.
47	Реализовать анализ влияния категориальных признаков на качество модели до и после кодирования.	3 — примеры; 4 — результаты модели; 3 — объяснение различий.
48	Исследовать влияние несбалансированных классов на деревья решений и случайный лес.	3 — реализация; 4 — сравнение; 3 — интерпретация.
49	Обучить модель классификации на произвольных текстовых данных (TF-IDF + логистическая регрессия).	3 — предобработка; 4 — обучение и метрики; 3 — выводы.
50	Проанализировать взаимосвязь между числом признаков и переобучением модели.	3 — изменение размерности; 4 — обучение и график; 3 — интерпретация результатов.

Деятельность студентов по формированию и развитию навыков учебной самостоятельной работы

В процессе самостоятельной работы студент приобретает навыки самоорганизации, самоконтроля, самоуправления, саморефлексии и становится активным самостоятельным субъектом учебной деятельности.

Выполняя самостоятельную работу под контролем преподавателя студент должен:

– освоить минимум содержания, выносимый на самостоятельную работу студентов и предложенный преподавателем в соответствии с образовательными стандартами высшего образования;

– планировать самостоятельную работу в соответствии с графиком самостоятельной работы, предложенным преподавателем;

– самостоятельную работу студент должен осуществлять в организационных формах, предусмотренных учебным планом и рабочей программой преподавателя;

– выполнять самостоятельную работу и отчитываться по ее результатам в соответствии с графиком представления результатов, видами и сроками отчетности по самостоятельной работе студентов.

студент может:

сверх предложенного преподавателем (при обосновании и согласовании с ним) и минимума обязательного содержания, определяемого программой по данной дисциплине:

– самостоятельно определять уровень (глубину) проработки содержания материала;

– предлагать темы и вопросы для самостоятельной проработки;

– в рамках общего графика выполнения самостоятельной работы предлагать обоснованный индивидуальный график выполнения и отчетности по результатам самостоятельной работы;

– предлагать свои варианты организационных форм самостоятельной работы;

– использовать для самостоятельной работы методические пособия, учебные пособия, разработки сверх предложенного преподавателем перечня;

– использовать не только контроль, но и самоконтроль результатов самостоятельной работы в соответствии с методами самоконтроля, предложенными преподавателем или выбранными самостоятельно.

Самостоятельная работа студентов должна оказывать важное влияние на формирование личности будущего бакалавра, она планируется студентом самостоятельно. Каждый студент самостоятельно определяет режим своей работы и меру труда, затрачиваемого на овладение учебным содержанием по каждой дисциплине. Он выполняет внеаудиторную работу по личному индивидуальному плану, в зависимости от его подготовки, времени и других условий.

Вопросы для проверки СРС

1. Что такое машинное обучение? Чем оно отличается от программирования по правилам?
2. Какие типы задач решаются с помощью обучения с учителем?
3. В чём отличие обучения с учителем от обучения без учителя?
4. Приведите примеры задач классификации и регрессии.
5. Что такое обучающая и тестовая выборка? Почему важно их разделять?
6. Что такое переобучение и как его обнаружить?
7. Какие методы позволяют бороться с переобучением?
8. В чём разница между L1- и L2-регуляризацией?
9. Что такое признаки (features) и целевая переменная (target)?
10. Зачем нужны методы нормализации и стандартизации признаков?
11. Что такое one-hot encoding и в каких случаях он применяется?
12. Что такое линейная регрессия и какие задачи она решает?
13. Как интерпретируются коэффициенты в модели линейной регрессии?
14. Что такое логистическая регрессия и чем она отличается от линейной?
15. Для каких задач подходит логистическая регрессия?
16. Как работает сигмоидная функция и как она используется в логистической регрессии?
17. Что такое точность (accuracy) классификации?

18. В чём отличие между precision и recall?
19. Когда важно использовать F1-меру?
20. Что показывает ROC-кривая и AUC?
21. Как работает метод ближайших соседей (KNN)?
22. Как выбрать оптимальное количество соседей в KNN?
23. Что такое метод опорных векторов (SVM)?
24. Чем отличаются линейное и нелинейное ядро в SVM?
25. Как работает стохастический градиентный спуск?
26. Зачем использовать мини-батчи при обучении модели?
27. Что такое дерево решений и как оно строится?
28. Как дерево решений принимает решение на новых данных?
29. Почему деревья склонны к переобучению и как это предотвратить?
30. В чём преимущества ансамблевых моделей по сравнению с одиночными?
31. Что такое случайный лес и как он работает?
32. Что такое градиентный бустинг?
33. В чём отличие K-средних от иерархической кластеризации?
34. Что такое PCA и зачем оно используется?
35. Чем отличается PCA от t-SNE?
36. Какие методы оценки качества кластеризации вы знаете?
37. Что такое гиперпараметры модели?
38. Как работает GridSearchCV и для чего он нужен?
39. Какие библиотеки Python используются для машинного обучения?
40. Как интерпретировать результаты модели и принять решение об их применимости?

ОБЩИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Основной формой самостоятельной работы студента является изучение конспекта лекций, их дополнение, рекомендованной литературы, активное участие на лабораторных занятиях. Но для успешной учебной деятельности, ее интенсификации, необходимо учитывать следующие субъективные факторы:

1. Знание школьного программного материала, наличие прочной системы зияний, необходимой для усвоения основных вузовских курсов. Это особенно важно для математических дисциплин. Необходимо отличать пробелы в знаниях, затрудняющие усвоение нового материала, от малых способностей. Затратив силы на преодоление этих пробелов, студент обеспечит себе нормальную успеваемость и поверит в свои способности.

2. Наличие умений, навыков умственного труда:

а) умение конспектировать на лекции и при работе с книгой;

б) владение логическими операциями: сравнение, анализ, синтез, обобщение, определение понятий, правила систематизации и классификации.

3. Специфика познавательных психических процессов: внимание, память, речь, наблюдательность, интеллект и мышление. Слабое развитие каждого из них становится серьезным препятствием в учебе.

4. Хорошая работоспособность, которая обеспечивается нормальным физическим состоянием. Ведь серьезное учение — это большой многосторонний и разнообразный труд. Результат обучения оценивается не количеством сообщаемой информации, а качеством ее усвоения, умением ее использовать и развитием у себя способности к дальнейшему самостоятельному образованию.

5. Соответствие избранной деятельности, профессии индивидуальным способностям. Необходимо выработать у себя умение саморегулировать свое

эмоциональное состояние и устранять обстоятельства, нарушающие деловой настрой, мешающие намеченной работе.

6. Овладение оптимальным стилем работы, обеспечивающим успех в деятельности. Чередование труда и пауз в работе, периоды отдыха, индивидуально обоснованная норма продолжительности сна, предпочтение вечерних или утренних занятий, стрессоустойчивость на экзаменах и особенности подготовки к ним,

7. Уровень требований к себе, определяемый сложившейся самооценкой.

Адекватная оценка знаний, достоинств, недостатков - важная составляющая самоорганизации человека, без нее невозможна успешная работа по управлению своим поведением, деятельностью.

Одна из основных особенностей обучения в высшей школе заключается в том, что постоянный внешний контроль заменяется самоконтролем, активная роль в обучении принадлежит уже не столько преподавателю, сколько студенту.

Зная основные методы научной организации умственного труда, можно при наименьших затратах времени, средств и трудовых усилий достичь наилучших результатов.

Эффективность усвоения поступающей информации зависит от работоспособности человека в тот или иной момент его деятельности.

Работоспособность - способность человека к труду с высокой степенью напряженности в течение определенного времени. Различают внутренние и внешние факторы работоспособности.

К внутренним факторам работоспособности относятся интеллектуальные особенности, воля, состояние здоровья.

К внешним:

- организация рабочего места, режим труда и отдыха;
- уровень организации труда - умение получить справку и пользоваться информацией;
- величина умственной нагрузки.

Выдающийся русский физиолог Н. Е. Введенский выделил следующие условия продуктивности умственной деятельности:

- во всякий труд нужно входить постепенно;
- мерность и ритм работы. Разным людям присущ более или менее разный темп работы;
- привычная последовательность и систематичность деятельности;
- правильное чередование труда и отдыха.

Отдых не предполагает обязательного полного бездействия со стороны человека, он может быть достигнут простой переменой дела. В течение дня работоспособность изменяется. Наиболее плодотворным является *утреннее время (с 8 до 14 часов),* причем максимальная работоспособность приходится на период с 10 до 13 часов, затем *послеобеденное* - (с 16 до 19 часов) и *вечернее* (с 20 до 24 часов). Очень трудный для понимания материал лучше изучать в начале каждого отрезка времени (лучше всего утреннего) после хорошего отдыха. Через 1-1,5 часа нужны перерывы по 10 - 15 мин, через 3 - 4 часа работы отдых должен быть продолжительным - около часа.

Составной частью научной организации умственного труда является овладение техникой умственного труда.

Время, которым располагает студент для выполнения учебного плана, складывается из двух составляющих: одна из них - это аудиторная работа в вузе по расписанию занятий, другая - внеаудиторная самостоятельная работа. Задания и материалы для самостоятельной работы выдаются во время учебных занятий по расписанию, на этих же занятиях преподаватель осуществляет контроль за самостоятельной работой, а также оказывает помощь студентам по правильной организации работы.

Самостоятельные занятия потребуют интенсивного умственного труда, который необходимо не только правильно организовать, но и стимулировать. При этом очень важно уметь поддерживать устойчивое внимание к изучаемому материалу. Выработка внимания требует значительных волевых усилий. Именно поэтому, если студент замечает, что он часто отвлекается во время самостоятельных занятий, ему надо заставить себя сосредоточиться. Подобную процедуру необходимо проделывать постоянно, так как это является тренировкой внимания. Устойчивое внимание появляется тогда, когда человек относится к делу с интересом.

Следует правильно организовать свои занятия по времени: 50 минут - работа, 5-10 минут - перерыв; после 3 часов работы перерыв - 20-25 минут. Иначе нарастающее утомление повлечет неустойчивость внимания. Очень существенным фактором, влияющим на повышение умственной работоспособности, являются систематические занятия физической культурой. Организация активного отдыха предусматривает чередование умственной и физической деятельности, что полностью восстанавливает работоспособность человека.

УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература:

Болотский, А. В.
    Исследование операций и методы оптимизации Электронный ресурс / Болотский А. В., Кочеткова О. А. - Санкт-Петербург : Лань, 2020. - 116 с. - ISBN 978-5-8114-4568-4

Сараев, П. В.
    Методы машинного обучения Электронный ресурс : Методические указания и задания к лабораторным работам по курсу / П. В. Сараев. - Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2017. - 48 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397

Сопов, Е. А.
    Многокритериальные нейроэволюционные системы в задачах машинного обучения и человеко-машинного взаимодействия Электронный ресурс / Сопов Е. А., Иванов И. А. : монография. - Красноярск : СФУ, 2019. - 160 с. - ISBN 978-5-7638-3969-2

Дополнительная литература:

Методические рекомендации к практическим занятиям по дисциплине "Теория и практика машинного перевода" : Направление подготовки 45.04.02 Лингвистика. Магистерская программа "Перевод и переводоведение".Квалификация выпускника - магистр. Форма обучения - очная. Учебный план 2015 года. Изучается в 3 семестре. - Ставрополь : СКФУ, 2015. - 9 с.

Неделько, В. М.
    Основы статистических методов машинного обучения Электронный ресурс : Учебное пособие / В. М. Неделько. - Новосибирск : Новосибирский государственный технический университет, 2010. - 72 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 978-5-7782-1385-2

Рапаков, Г. Г.
    Методы и алгоритмы машинного обучения при принятии управленческих решений в региональной системе медицинской профилактики (опыт Вологодской области) Электронный ресурс / Рапаков Г. Г., Касимов Р. А. : монография. - Вологда : ВоГУ, 2014. - 143 с. - ISBN 978-5-87851-548-1

Учебно-методическая литература:

Сараев, П. В.
    Методы машинного обучения Электронный ресурс : Методические указания и задания к лабораторным работам по курсу / П. В. Сараев. - Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2017. - 48 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397