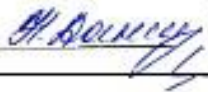


Муниципальное автономное общеобразовательное учреждение города
Новосибирска
«Лицей №22 «Надежда Сибири»
Главный корпус на Советской: г. Новосибирск, ул. Советская, 63, тел. 222-35-15,
e-mail: l_22@edu54.ru
Корпус 99 на Чаплыгина: г. Новосибирск, ул. Чаплыгина, 59, тел. 223-74-15

РАССМОТРЕНО на заседании кафедры математического образования, протокол № 1 от 19.08.2025  Максупова С.Н.	СОГЛАСОВАНО Протокол № 3 от 29.08.2025 Заместитель директора  Н.А. Данилова
--	---

РАБОЧАЯ ПРОГРАММА

Информатика. Основы технологий искусственного интеллекта

10-11 ИП

(уровень среднего общего образования)

Разработчик:

Кириленко Юрий Александрович

Рабочая программа по учебному предмету «Информатика. Искусственный интеллект» (предметная область «Математика и информатика») (далее соответственно – программа по искусственному интеллекту, искусственный интеллект) составлена на основе Федеральной рабочей программы по информатике и является авторской, включает пояснительную записку, содержание обучения, планируемые результаты освоения программы по искусственному интеллекту, тематическое планирование.

Пояснительная записка отражает общие цели и задачи изучения искусственного интеллекта, место в структуре учебного плана, а также подходы к отбору содержания, к определению планируемых результатов.

Содержание обучения раскрывает содержательные линии, которые предлагаются для обязательного изучения в 10 классе на уровне среднего общего образования.

Планируемые результаты освоения программы по искусственному интеллекту включают личностные, метапредметные результаты за период обучения в 10 классе на уровне среднего общего образования, а также предметные достижения обучающегося.

1. Пояснительная записка

Общая характеристика учебного предмета

Искусственный интеллект, или наука о данных, представляет собой междисциплинарную область, объединяющую статистику, математику и информатику для анализа данных и извлечения из них полезной информации. Обучение основам Искусственный интеллект в школе открывает учащимся двери в мир анализа данных, помогая им развивать критическое мышление и навыки решения проблем, которые востребованы в современном цифровом обществе.

Функциональная значимость Искусственного интеллекта для школьников заключается в его способности обеспечивать понимание методов сбора, обработки и анализа данных. Эти знания необходимы для принятия обоснованных решений и решения разнообразных задач в различных сферах, включая науку, бизнес, медицину и социальные исследования.

Знание Искусственного интеллекта и умение работать с данными важно для каждого школьника, стремящегося понять современные технологии и их применение в реальной жизни. Понимание основ статистики, программирования и анализа данных помогает учащимся развивать аналитическое мышление, способности к исследовательской деятельности и креативность.

Искусственный интеллект, выполняя свои основные функции, позволяет учащимся работать с большими объемами данных, визуализировать результаты анализа, строить модели и делать предсказания. Эти навыки необходимы для успешного обучения и будущей профессиональной деятельности в различных областях.

Обучение Искусственному интеллекту направлено на развитие интеллектуальных и технических способностей учащихся, включая умение работать с данными, использовать инструменты анализа и визуализации, понимать и применять статистические методы. Это способствует развитию аналитического и

логического мышления, навыков программирования и самообразования, что является важным для успешного обучения и дальнейшего профессионального роста.

Содержание программы по Искусственному интеллекту ориентировано также на развитие функциональной грамотности учащихся, включая умение читать и понимать научные тексты, использовать информацию из различных источников, оценивать ее и применять на практике для достижения поставленных целей, расширения своих знаний и возможностей, участия в социальной и профессиональной жизни.

Цели и задачи изучения учебного предмета «Искусственный интеллект».

Изучение искусственного интеллекта направлено на достижение следующих целей:

- Развитие аналитических навыков: Ознакомление школьников с основами Data Science для развития навыков анализа данных и принятия обоснованных решений.
- Повышение интереса к науке и технологиям: Стимулирование интереса учащихся к современным технологиям и методам обработки данных через практическое применение полученных знаний.
- Подготовка к профессиональной деятельности: Формирование базовых навыков, необходимых для дальнейшего обучения и профессиональной деятельности в сфере Искусственного интеллекта и смежных областях.

Задачи:

- Изучение основ Искусственного интеллекта: Ознакомление учащихся с основами работы с данными, включая сбор, обработку и анализ данных.
- Практическое применение знаний: Создание учебных проектов, позволяющих закрепить знания на практике, включая анализ реальных данных и визуализацию результатов.
- Работа с инструментами Искусственного интеллекта: Обучение использованию инструментов и языков программирования, таких как Python, для анализа данных, работы с библиотеками NumPy, pandas, matplotlib и scikit-learn.
- Применение статистических методов: Ознакомление с основными статистическими методами и их применением для анализа данных и построения моделей.
- Работа с большими данными: Ознакомление с основами работы с большими объемами данных и методами их обработки и анализа.
- Развитие навыков самостоятельной работы: Поощрение самостоятельного изучения новых инструментов и методов анализа данных, поиска решений и выполнения творческих проектов.
- Оценка и использование информации: Развитие у учащихся умений критически оценивать и использовать информацию из различных источников для решения поставленных задач.

- Социальная и профессиональная активность: Стимулирование участия учащихся в учебных и внеучебных проектах, конкурсах и олимпиадах, направленных на анализ данных и решение реальных проблем.

Особенности классов

Рабочая программа по предмету «Информатика. Искусственный интеллект» для 10-го и 11-го «ИП» класса предназначена для углубленного изучения учащимися информационно-технологического профиля в группе «Программисты». На изучение данного модуля отведено 66 часов в 10-м классе и 60 часов в 11 классе.

Место предмета в учебном плане лицея

Учебный план на изучение «Информатика. Искусственный интеллект» в 10 ПИ, 10 «И» классе среднего общего образования отводит 2 учебный час в неделю (всего 30 учебных часов) за счёт части, формируемой участниками образовательных отношений.

Учебный год	Количество часов
	10 «ИП»
2025/2026	66
2026/2027	60

К тематическому планированию применяется модульный принцип построения образовательной программы, что позволяет выстраивать индивидуальную образовательную парадигму и обеспечивать саморазвитие при индивидуальном темпе работы с учебным материалом, контроль и самоконтроль знаний.

Используемые образовательные технологии, в том числе дистанционные

Обучение искусственному интеллекту может осуществляться с использованием дистанционных образовательных технологий (далее ДОТ), которое предполагает как самостоятельное прохождение учебного материала учеником, так и с помощью сопровождения учителя. При применении ДОТ используются платформы: лицейская платформа дистанционного обучения Moodle, ФГИС «Моя школа», ГИС «Электронная школа» Новосибирской области.

При реализации рабочей программы могут быть использованы материалы для подготовки к профилям олимпиады КД НТИ и стандартов Всероссийского чемпионатного движения по профессиональному мастерству «Профессионалы».

Информация о промежуточной аттестации

Промежуточная аттестация осуществляется по окончании учебного модуля с целью проверки степени и качества усвоения материала по результатам изучения тематических модулей и проводится в форме аттестационных работ.

Текущий контроль осуществляются с целью проверки степени и качества усвоения материала в ходе его изучения в следующих формах: самостоятельных и проверочных работ.

Текущий контроль и промежуточная аттестация осуществляются в соответствии с «Положением об осуществлении текущего контроля успеваемости и промежуточной аттестации обучающихся, их формах, периодичности и порядке проведения муниципального автономного общеобразовательного учреждения города

Новосибирска «Лицей № 22 «Надежда Сибири» (протокол педагогического совета №1 от 29.08.2023).

Итоговая аттестация проводится в соответствии с законодательством РФ.

**Промежуточная аттестация
по искусственному интеллекту в 10 «ИП»**

№ модульн ой	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
МР № 1	Предобработка данных	18	18	Аттестацио нная работа
МР № 2	Введение в машинное обучение	16	34	Аттестацио нная работа
МР № 3	Методы оптимизации	16	50	Аттестацио нная работа
МР № 4	Введение в нейронные сети	16	66	Аттестацио нная работа

**Промежуточная аттестация
по искусственному интеллекту в 11 «ИП»**

№ модульн ой	Название модуля	Количество часов в модуле	Номер урока ПА	Форма ПА
МР № 1	Нейронные сети	20	20	Аттестацио нная работа
МР № 2	Основы обработки естественного языка	22	42	Аттестацио нная работа
МР № 3	Временные ряды	18	60	Аттестацио нная работа

2. Содержание учебного предмета

10 класс

Модуль 1 «Предобработка данных»

Освоение математических основ Python, включая работу с библиотеками NumPy и SciPy для выполнения сложных вычислений, операций линейной алгебры и математического моделирования. Практическая работа с табличными данными через библиотеку Pandas: загрузка, очистка, преобразование и анализ структурированных данных, использование группировок, сводных таблиц и методов обработки пропущенных значений. Обработка текстовой информации с применением регулярных выражений, методов NLP для токенизации, лемматизации и анализа тональности текста. Работа со структурированными данными: парсинг JSON и XML, чтение и запись CSV/TSV файлов, преобразование между форматами. Основы цифровой обработки изображений с использованием OpenCV и PIL: фильтрация, преобразование, сегментация и анализ изображений. Визуализация данных средствами Matplotlib и Seaborn: построение графиков, гистограмм, диаграмм рассеяния и сложных композиций. Картографическая визуализация через API картографических сервисов и библиотеку Folium для создания интерактивных карт. Использование инструментов визуализации таблиц (например, Tabulate) для презентации данных. Методы предобработки данных: масштабирование признаков, кодирование категориальных переменных, обработка выбросов и дисбаланса классов. Завершение модуля практической работой, интегрирующей все изученные методы для решения комплексной задачи анализа и визуализации реальных данных.

Модуль 2 «Введение в машинное обучение»

Освоение введения в машинное обучение, включая базовые понятия, типы задач (обучение с учителем, без учителя, с подкреплением) и основные этапы workflow. Изучение регрессионного анализа моделей: от простой линейной регрессии до множественной полиномиальной, оценка качества моделей через метрики MSE, MAE, R^2 . Практическая работа с ансамблевыми методами на основе деревьев решений — случайный лес и градиентный бустинг (XGBoost, LightGBM), их настройка и применение для задач классификации и регрессии. Исследование метода опорных векторов (SVM) для линейной и нелинейной классификации, подбор гиперпараметров и ядерных функций. Освоение байесовских методов классификации, включая наивный байесовский классификатор и его применение к текстовым данным. Изучение алгоритмов кластеризации данных — k-средних, DBSCAN, иерархической кластеризации — для выявления скрытых паттернов и сегментации данных. Завершение модуля практической работой, интегрирующей все изученные методы: от предобработки данных и построения моделей до их оценки и интерпретации результатов, с акцентом на решение реальных задач машинного обучения.

Модуль 3 «Методы оптимизации»

Освоение бэггинг-ансамблей моделей, включая алгоритм случайного леса как ключевой пример, с акцентом на снижение дисперсии и повышение устойчивости моделей через агрегацию предсказаний множества независимых базовых алгоритмов. Изучение бустинг-ансамблей, таких как AdaBoost, Gradient Boosting и XGBoost, с

фокусом на последовательное улучшение предсказаний за счет коррекции ошибок предыдущих моделей и минимизацию смещения. Практическая работа со стекинг-ансамблями, комбинирующими разнородные модели через мета-обучатель для достижения более точных и robust предсказаний. Применение матричных разложений: метода главных компонент (PCA) для снижения размерности и визуализации данных, а также независимого компонентного анализа (ICA) для разделения смешанных сигналов. Освоение обучения метрических преобразований, включая метод k ближайших соседей с обучением метрики (LMNN) для улучшения классификации в пространстве признаков. Исследование методов подбора гиперпараметров: от сеточного и случайного поиска до оптимизации на основе байесовских методов и оптимизации в режиме кросс-валидации. Завершение модуля практической работой, интегрирующей все изученные методы оптимизации: от выбора и настройки ансамблевых моделей до применения методов снижения размерности и тонкой настройки гиперпараметров для достижения максимальной производительности моделей на реальных данных.

Модуль 4 «Введение в нейронные сети»

Освоение архитектуры полносвязных нейронных сетей, включая понимание слоев, функций активации и принципов forward propagation. Изучение процесса обучения нейронных сетей через алгоритм обратного распространения ошибки (backpropagation), расчет градиентов и обновление весов с использованием методов оптимизации (SGD, Adam). Практическое введение в фреймворк PyTorch: создание тензоров, построение вычислительных графов, определение архитектуры моделей через модуль nn и организация циклов обучения. Исследование методов ускорения обучения (инициализация весов, изменение learning rate) и снижения переобучения (регуляризация L1/L2, dropout, batch normalization). Освоение основ компьютерного зрения: обработка изображений, сверточные операции и знакомство с простыми CNN-архитектурами для задач классификации изображений. Завершение модуля практической работой, интегрирующей все изученные концепции: от проектирования и обучения полносвязных сетей до применения методов борьбы с переобучением и решения базовых задач компьютерного зрения с использованием PyTorch.

11 класс

Модуль 1 «Нейронные сети»

Освоение повторения материалов по архитектурам нейронных сетей, включая закрепление понимания прямого и обратного распространения, а также принципов работы сверточных слоев. Изучение различных видов сверток: стандартных, расширенных (dilated), разделяемых (depthwise) и транспонированных, их применения для эффективного извлечения признаков из многомерных данных. Практическая работа с методами оптимизации нейронных сетей: адаптивные алгоритмы (Adam, RMSprop), планировщики скорости обучения, методы ускорения сходимости и избежания локальных минимумов. Исследование современных методов регуляризации: L1/L2-регуляризация, dropout, batch normalization, label smoothing и их роль в снижении переобучения сложных моделей. Освоение техник аугментации данных для изображений, текстов и временных рядов, позволяющих увеличивать

разнообразие тренировочных данных без сбора новых образцов. Изучение переноса обучения и работы с предобученными моделями, создание и использование векторных представлений изображений для решения задач с ограниченными данными. Практическое применение методов детекции и локализации объектов на основе архитектуры YOLO (You Only Look Once) для реального времени обработки видео. Освоение семантической сегментации изображений с использованием U-Net и подобных архитектур для поточечной классификации. Завершение модуля практической работой, интегрирующей все изученные методы: от построения и оптимизации сложных сверточных сетей до решения практических задач компьютерного зрения с применением переноса обучения, аугментации и современных архитектур для детекции и сегментации.

Модуль 2 «Основы обработки естественного языка»

Освоение методов создания векторных представлений текста, включая изучение подходов TF-IDF, Bag of Words и современных эмбединговых техник. Практическая работа по предобработке текстовых данных: токенизация, лемматизация, стемминг, удаление стоп-слов и обработка специальных символов для подготовки корпусов текстов. Изучение алгоритма word2vec и его архитектур (Skip-gram и CBOW) для обучения распределенных векторных представлений слов, сохраняющих семантические и синтаксические отношения. Выполнение практической работы по обучению и визуализации word2vec-моделей на реальных текстовых данных с анализом полученных эмбедингов. Освоение архитектур рекуррентных нейронных сетей (RNN), включая LSTM и GRU, для обработки последовательностей текста и учета контекстной зависимости. Практическая работа по построению и обучению RNN-моделей для задач классификации текстов, генерации последовательностей и анализа тональности. Завершение модуля итоговой работой, интегрирующей все изученные методы обработки естественного языка: от предобработки сырых текстовых данных и создания векторных представлений до построения сложных рекуррентных сетей для решения практических задач NLP с оценкой качества моделей и интерпретацией результатов.

Модуль 3 «Временные ряды»

Освоение основ анализа временных рядов, включая изучение компонент ряда (тренд, сезонность, шум), стационарности и методов их визуализации. Практическое применение методов машинного обучения для прогнозирования временных рядов: от линейных моделей и случайного леса до градиентного бустинга и рекуррентных нейронных сетей. Изучение библиотеки TSNA (Time Series Numerical Analysis) для вычисления статистических характеристик, декомпозиции рядов и выявления аномалий. Освоение адаптивных методов прогнозирования, таких как экспоненциальное сглаживание (ETS), модель Хольта-Винтерса и ARIMA-семейство моделей, включая подбор параметров и оценку качества прогнозов. Завершение модуля практической работой, интегрирующей все изученные методы: от предобработки и анализа временных рядов до

построения и сравнения различных моделей прогнозирования с интерпретацией результатов и оценкой их точности на реальных данных.

3. Планируемые образовательные результаты освоения учебного предмета Искусственный интеллект

Личностные результаты

1. сформированность мировоззрения, соответствующего современному уровню развития науки и техники;
2. готовность и способность к образованию, в том числе самообразованию, на протяжении всей жизни; сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности;
3. навыки сотрудничества со сверстниками, детьми младшего возраста, взрослыми в образовательной, учебно-исследовательской, проектной и других видах деятельности;
4. эстетическое отношение к миру, включая эстетику научного и технического творчества;
5. осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов; отношение к профессиональной деятельности как возможности участия в решении личных, общественных, государственных, общенациональных проблем.

Метапредметные результаты:

1.2.5	Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях
1.2.6	Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду
1.2.7	Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учетом анализа имеющихся материальных и нематериальных ресурсов
1.3	Работа с информацией
1.3.1	Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления
1.3.2	Создавать тексты в различных форматах с учетом назначения информации и целевой аудитории, выбирая оптимальную форму представления и визуализации
1.3.3	Оценивать достоверность, легитимность информации, ее соответствие правовым и морально-этическим нормам
1.3.4	Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности
1.3.5	Владеть навыками распознавания и защиты информации, информационной безопасности личности
2	Коммуникативные УУД
2.1	Общение
2.1.1	Осуществлять коммуникации во всех сферах жизни; владеть различными способами общения и взаимодействия
2.1.2	Развернуто и логично излагать свою точку зрения с использованием языковых средств
2.1.3	Аргументированно вести диалог
3	
3.1	Самоорганизация
3.1.1	Самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; давать оценку новым ситуациям
3.1.2	Самостоятельно составлять план решения проблемы с учетом имеющихся ресурсов, собственных возможностей и предпочтений; делать осознанный выбор, аргументировать его, брать ответственность за решение; оценивать приобретенный опыт; способствовать формированию и проявлению широкой эрудиции в разных областях знаний
3.2	Самоконтроль

3.2.1	Давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям
3.2.2	Владеть навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и оснований; использовать приемы рефлексии для оценки ситуации, выбора верного решения; уметь оценивать риски и своевременно принимать решения по их снижению
3.3	Эмоциональный интеллект, предполагающий сформированность: саморегулирования, включающего самоконтроль, умение принимать ответственность за свое поведение, способность адаптироваться к эмоциональным изменениям и проявлять гибкость, быть открытым новому; внутренней мотивации, включающей стремление к достижению цели и успеху, оптимизм, инициативность, умение действовать, исходя из своих возможностей

Выпускник научится:

1. самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;

2. продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников деятельности, эффективно разрешать конфликты;

3. владеть навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем;

4. использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.

Выпускник получит возможность научиться:

1. быть готовым и способным к самостоятельной информационно-познавательной деятельности, включая умение ориентироваться в различных источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников;

2. быть способным и готовым к самостоятельному поиску методов решения практических задач, применению различных методов познания.

Регулятивные универсальные учебные действия

1. самостоятельно определять цели, задавать параметры и критерии, по которым можно определить, что цель достигнута;

2. оценивать возможные последствия достижения поставленной цели в деятельности, собственной жизни и жизни окружающих людей, основываясь на соображениях этики и морали;

3. ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях;

4. оценивать ресурсы, в том числе время и другие нематериальные ресурсы, необходимые для достижения поставленной цели;

5. выбирать путь достижения цели, планировать решение поставленных задач, оптимизируя материальные и нематериальные затраты;

6. организовывать эффективный поиск ресурсов, необходимых для достижения поставленной цели;

7. сопоставлять полученный результат деятельности с поставленной заранее целью.

Познавательные универсальные учебные действия

1. искать и находить обобщенные способы решения задач, в том числе, осуществлять развернутый информационный поиск и ставить на его основе новые (учебные и познавательные) задачи;

2. критически оценивать и интерпретировать информацию с разных позиций, распознавать и фиксировать противоречия в информационных источниках;

3. использовать различные модельно-схематические средства для представления существенных связей и отношений, а также противоречий, выявленных в информационных источниках;

4. находить и приводить критические аргументы в отношении действий и суждений другого; спокойно и разумно относиться к критическим замечаниям в отношении собственного суждения, рассматривать их как ресурс собственного развития;

5. выходить за рамки учебного предмета и осуществлять целенаправленный поиск возможностей для широкого переноса средств и способов действия;

6. выстраивать индивидуальную образовательную траекторию, учитывая ограничения со стороны других участников и ресурсные ограничения;

7. менять и удерживать разные позиции в познавательной деятельности.

Коммуникативные универсальные учебные действия

1. осуществлять деловую коммуникацию как со сверстниками, так и со взрослыми (как внутри образовательной организации, так и за ее пределами), подбирать партнеров для деловой коммуникации исходя из соображений результативности взаимодействия, а не личных симпатий;

2. при осуществлении групповой работы быть как руководителем, так и членом команды в разных ролях (генератор идей, критик, исполнитель, выступающий, эксперт и т.д.);

3. координировать и выполнять работу в условиях реального, виртуального и комбинированного взаимодействия;

4. развернуто, логично и точно излагать свою точку зрения с использованием адекватных (устных и письменных) языковых средств;

5. распознавать конфликтогенные ситуации и предотвращать конфликты до их активной фазы, выстраивать деловую и образовательную коммуникацию, избегая личностных оценочных суждений.

Предметные результаты

Выпускник будет демонстрировать:

1) систематизация знаний, относящихся к *математическим объектам информатики*; умение строить математические объекты информатики, в том числе логические формулы;

2) сформированность базовых навыков и умений по соблюдению требований *техники безопасности*, гигиены и ресурсосбережения при работе со средствами информатизации;

3) владение опытом построения и использования *компьютерно-математических моделей*, проведения экспериментов и статистической обработки данных с помощью компьютера, интерпретации результатов, получаемых в ходе моделирования реальных процессов; умение оценивать числовые параметры моделируемых объектов и процессов; сформированность представлений о необходимости *анализа соответствия модели* и моделируемого объекта (процесса);

4) сформированность представлений о способах хранения и простейшей обработке данных; умение пользоваться *базами данных* и справочными системами; владение основными сведениями о базах данных, их структуре, средствах создания и работы с ними;

Выпускник получит возможность продемонстрировать:

1. владение системой базовых знаний, отражающих *вклад информатики* в формирование современной научной картины мира;

2. владение навыками *алгоритмического мышления* и понимание необходимости формального описания алгоритмов.

ПРОВЕРЯЕМЫЕ НА ЕГЭ ПО ИНФОРМАТИКЕ ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ СРЕДНЕГО ОБЩЕГО ОБРАЗОВАНИЯ

Код проверяемого требования	Проверяемые требования к предметным результатам освоения основной образовательной программы среднего общего образования
1.	<i>Знать (понимать)</i>
1.1	Понимание основных принципов устройства и функционирования современных стационарных и мобильных компьютеров; тенденций развития компьютерных технологий; владение навыками работы с операционными системами и основными видами программного обеспечения для решения учебных задач по выбранной специализации
1.4	Понимание базовых алгоритмов обработки числовой и текстовой информации (запись чисел в позиционной системе счисления, делимость целых чисел; нахождение всех простых чисел в

Код проверяемого требования	Проверяемые требования к предметным результатам освоения основной образовательной программы среднего общего образования
	заданном диапазоне; обработка многоразрядных целых чисел; анализ символьных строк и других), алгоритмов поиска и сортировки
1.5	Знание функциональные возможности инструментальных средств среды разработки
1.6	Владение основными сведениями о базах данных, их структуре, средствах создания и работы с ними
1.7	Понимание возможностей и ограничений технологий искусственного интеллекта в различных областях; наличие представлений об использовании информационных технологий в различных профессиональных сферах
2.	<i>Уметь</i>
2.1	Умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде
2.2	Умение классифицировать основные задачи анализа данных (прогнозирование, классификация, кластеризация, анализ отклонений); понимать последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и (или) построение модели, преобразование данных, визуализация данных, интерпретация результатов
2.9	Умение анализировать алгоритмы с использованием таблиц трассировки; определять без использования компьютера результаты выполнения несложных программ, включающих циклы, ветвления и подпрограммы, при заданных исходных данных
2.10	Умение определять сложность изучаемых в курсе базовых алгоритмов (суммирование элементов массива, сортировка массива, переборные алгоритмы, двоичный поиск) и приводить примеры нескольких алгоритмов разной сложности для решения одной задачи
2.11	Владение универсальным языком программирования высокого уровня (Паскаль, Python, Java, C++, C#), представлениями о базовых типах данных и структурах данных; умение использовать основные управляющие конструкции; умение осуществлять анализ предложенной программы: определять

Код проверяемого требования	Проверяемые требования к предметным результатам освоения основной образовательной программы среднего общего образования
	результаты работы программы при заданных исходных данных; определять, при каких исходных данных возможно получение указанных результатов; выявлять данные, которые могут привести к ошибке в работе программы; формулировать предложения по улучшению программного кода
2.12	Умение реализовывать на выбранном для изучения языке программирования высокого уровня (Паскаль, Python, Java, C++, C#) типовые алгоритмы обработки чисел, числовых последовательностей и массивов: представление числа в виде набора простых сомножителей; нахождение максимальной (минимальной) цифры натурального числа, записанного в системе счисления с основанием, не превышающим 10; вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию); сортировку элементов массива; умение использовать в программах данные различных типов с учётом ограничений на диапазон их возможных значений, применять при решении задач структуры данных (списки, словари, стеки, очереди, деревья); применять стандартные и собственные подпрограммы для обработки числовых данных и символьных строк; использовать при разработке программ библиотеки подпрограмм; умение использовать средства отладки программ в среде программирования

ПЕРЕЧЕНЬ ЭЛЕМЕНТОВ СОДЕРЖАНИЯ, ПРОВЕРЯЕМЫХ НА ЕГЭ ПО ИНФОРМАТИКЕ

Код	Проверяемый элемент содержания
3	Алгоритмы и программирование
3.1	Формализация понятия алгоритма. Машина Тьюринга как универсальная модель вычислений
3.2	Оценка сложности вычислений. Время работы и объём используемой памяти, их зависимость от размера исходных данных. Оценка асимптотической сложности алгоритмов. Алгоритмы полиномиальной сложности. Переборные алгоритмы. Примеры различных алгоритмов решения одной задачи, которые имеют различную сложность

Код	Проверяемый элемент содержания
3.3	Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат
3.4	Алгоритмы обработки натуральных чисел, записанных в позиционных системах счисления: разбиение записи числа на отдельные цифры, нахождение суммы и произведения цифр, нахождение максимальной (минимальной) цифры. Представление числа в виде набора простых сомножителей. Алгоритм быстрого возведения в степень. Поиск простых чисел в заданном диапазоне с помощью алгоритма «решето Эратосфена»
3.5	Многоразрядные целые числа, задачи длинной арифметики
3.6	Язык программирования (Паскаль, Python, Java, C++, C#). Типы данных: целочисленные, вещественные, символьные, логические. Ветвления. Сложные условия. Циклы с условием. Циклы по переменной. Обработка данных, хранящихся в файлах. Текстовые и двоичные файлы. Файловые переменные (файловые указатели). Чтение из файла. Запись в файл. Разбиение задачи на подзадачи. Подпрограммы (процедуры и функции). Использование стандартной библиотеки языка программирования
3.7	Рекурсия. Рекурсивные процедуры и функции. Использование стека для организации рекурсивных вызовов
3.8	Численные методы. Точное и приближённое решения задачи. Численное решение уравнений с помощью подбора параметра. Численные методы решения уравнений: метод перебора, метод половинного деления. Приближённое вычисление длин кривых. Вычисление площадей фигур с помощью численных методов (метод прямоугольников, метод трапеций). Поиск максимума (минимума) функции одной переменной методом половинного деления
3.9	Обработка символьных данных. Встроенные функции языка программирования для обработки символьных строк. Алгоритмы обработки символьных строк: подсчёт количества появлений символа в строке, разбиение строки на слова по пробельным символам, поиск подстроки внутри данной строки, замена найденной подстроки на другую строку. Генерация всех слов в некотором алфавите, удовлетворяющих заданным ограничениям. Преобразование числа в символьную строку и обратно
3.10	Массивы и последовательности чисел. Вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения, среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих

Код	Проверяемый элемент содержания
	заданному условию). Линейный поиск заданного значения в массиве. Алгоритмы работы с элементами массива с однократным просмотром массива. Сортировка одномерного массива. Простые методы сортировки (метод пузырька, метод выбора, сортировка вставками). Сортировка слиянием. Быстрая сортировка массива (алгоритм QuickSort). Двоичный поиск в отсортированном массиве
3.11	Двумерные массивы (матрицы). Алгоритмы обработки двумерных массивов: заполнение двумерного числового массива по заданным правилам, поиск элемента в двумерном массиве, вычисление максимума (минимума) и суммы элементов двумерного массива, перестановка строк и столбцов двумерного массива
3.12	Словари (ассоциативные массивы, отображения). Хэш-таблицы. Построение алфавитно-частотного словаря для заданного текста
3.13	Стеки. Анализ правильности скобочного выражения. Вычисление арифметического выражения, записанного в постфиксной форме. Очереди. Использование очереди для временного хранения данных
3.14	Алгоритмы на графах. Построение минимального остовного дерева взвешенного связного неориентированного графа. Количество различных путей между вершинами ориентированного ациклического графа. Алгоритм Дейкстры
3.15	Деревья. Реализация дерева с помощью ссылочных структур. Двоичные (бинарные) деревья. Построение дерева для заданного арифметического выражения. Рекурсивные алгоритмы обхода дерева. Использование стека и очереди для обхода дерева
3.16	Динамическое программирование как метод решения задач с сохранением промежуточных результатов. Задачи, решаемые с помощью динамического программирования: вычисление рекурсивных функций, подсчёт количества вариантов, задачи оптимизации
3.17	Понятие об объектно-ориентированном программировании. Объекты и классы. Свойства и методы объектов. Объектно-ориентированный анализ. Разработка программ на основе объектно-ориентированного подхода. Инкапсуляция, наследование, полиморфизм
4	Информационные технологии
4.1	Анализ данных. Основные задачи анализа данных: прогнозирование, классификация, кластеризация, анализ отклонений. Последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и (или) построение модели, преобразование данных, визуализация данных, интерпретация результатов. Программные средства и Интернет-сервисы для обработки и представления данных. Большие данные. Машинное обучение

Код	Проверяемый элемент содержания
4.2	Анализ данных с помощью электронных таблиц. Вычисление суммы, среднего арифметического, наибольшего (наименьшего) значения диапазона. Вычисление коэффициента корреляции двух рядов данных. Построение столбчатых, линейчатых и круговых диаграмм. Построение графиков функций. Подбор линии тренда, решение задач прогнозирования. Решение задач оптимизации с помощью электронных таблиц

4. Тематическое планирование

10 класс

10 класс					
№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрол ьные работы	Практич еские работы	
Модуль «Предобработка данных» - 18 часов					
1.1	Фундаментальная обработка и анализ табличных данных	9			https://stepik.org/course/135003/promo?search=7691628872
1.2	Работа со специализирован ными данными и визуализация	8			
1.3	Модульная работа «Предобработка данных»	1			
Модуль «Введение в машинное обучение» - 16 часов					
2.1	Обучение с учителем: от регрессии до классификации	8			https://stepik.org/course/135003/promo?search=7691628872
2.2	Обучение без учителя и ансамблевые методы	7			
2.3	Модульная работа «Введение в машинное обучение»	1			
Модуль «Методы оптимизации» - 16 часов					

3.1	Ансамблевые методы и композиции моделей	8			https://stepik.org/course/135003/promo?search=7691628872
3.2	Снижение размерности и оптимизация моделей	7			
3.4	Модульная работа «Методы оптимизации»	1			
Модуль «Введение в Нейронные сети» - 16 часов					
4.1	Основы нейронных сетей и фреймворки	8			https://stepik.org/course/135003/promo?search=7691628872
4.2	Практика глубокого обучения и компьютерное зрение	7			
4.3	Модульная работа «Введение в нейронные сети»	1			

11 класс

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрол ьные работы	Практич еские работы	
Модуль «Нейронные Сети» - 20 часов					
1.1	Архитектуры и методы оптимизации сверточных сетей	10			https://stepik.org/course/135003/promo?search=7691628872
1.2	Современные задачи компьютерного зрения и трансферное обучение	9			

1.3	Модульная работа «Предобработка данных»	1			
Модуль «Основы обработки естественного языка» - 22 часа					
2.1	Векторные представления и семантический анализ текста	11			https://stepik.org/course/135003/promo?search=7691628872
2.2	Рекуррентные нейронные сети и их применение	10			
2.3	Модульная работа «Введение в машинное обучение»	1			
Модуль «Временные ряды» - 18 часов					
3.1	Классические методы анализа и прогнозирования временных рядов	8			https://stepik.org/course/135003/promo?search=7691628872
3.2	Современные ML-подходы и адаптивное прогнозирование	9			
3.4	Модульная работа «Методы оптимизации»	1			

5. Приложения к программе

ПОУРОЧНОЕ ПЛАНИРОВАНИЕ

10 класс

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрол ьные работы	Практич еские работы	
Модуль «Предобработка данных» - 18 часов					
1.1	Математические основы Python	2			

1.2	Работа с табличными данными	2			https://stepik.org/course/135003/promo?search=7691628872
1.3	Обработка текстовой информации	2			
1.4	Структурированные данные: JSON и XML CSV TSV	2			
1.5	Цифровая обработка изображений	2			
1.6	Визуализация данных. Matplotlib	2			
1.7	Картографическая визуализация API	2			
1.8	Инструменты визуализации таблиц	2			
1.9	MP Предобработка данных	2			
Модуль «Введение в машинное обучение» - 16 часов					
2.1	Введение в машинное обучение	2			https://stepik.org/course/135003/promo?search=7691628872
2.2	Регрессионный анализ моделей	2			
2.3	Ансамбли на основе деревьев решений	2			
2.4	Метод опорных векторов (SVM)	2			
2.5	Байесовские методы классификации	2			
2.6	Алгоритмы кластеризации данных	4			
2.7	MP Введение в машинное обучение	2			

Модуль «Методы оптимизации» - 16 часов					
3.1	Бэггинг-ансамбли моделей	2			https://stepik.org/course/135003/promo?search=7691628872
3.2	Ансамбли: Бустинг	2			
3.4	Ансамбли: Стекинг	2			
3.5	Матричные разложения: PCA и ICA	2			
3.6	Обучение метрических преобразований	2			
3.7	Подбор гиперпараметров	4			
3.8	MP Методы оптимизации	2			
Модуль «Введение в Нейронные сети» - 16 часов					
4.1	Полносвязные нейронные сети	2			https://stepik.org/course/135003/promo?search=7691628872
4.2	Обучение нейронных сетей	2			
4.3	Введение в PyTorch	4			
4.4	Ускорение обучения и снижение переобучения	2			
4.5	Основы компьютерного зрения	4			
4.6	MP Введение в Нейронные сети	2			

11 класс

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрол ьные работы	Практи ческие работы	
Модуль «Нейронные Сети» - 20 часов					

1.1	Повторение материалов	4			https://stepik.org/course/135003/promo?search=7691628872
1.2	Виды сверток	2			
1.3	Методы оптимизации	2			
1.4	Регуляризация	2			
1.5	Аугментация	2			
1.6	Перенос обучения. Векторные представления изображений	2			
1.7	Детекция и локализация объектов YOLO	2			
1.8	Семантическая сегментация	2			
1.9	МР Нейронные Сети	2			
Модуль «Основы обработки естественного языка» - 22 часа					
2.1	Векторные представления текста	2			https://stepik.org/course/135003/promo?search=7691628872
2.2	Предобработка текстовых данных	4			
2.3	word2vec	2			
2.4	Практическая работа word2vec	4			
2.5	Рекуррентные нейронные сети	4			
2.6	Практическая работа RNN	4			
2.7	МР Основы обработки естественного языка	2			
Модуль «Временные ряды» - 18 часов					
3.1	Анализ временных рядов	4			https://stepik.org/course/135003/promo?search=7691628872
3.2	Машинное обучение для	4			

	прогнозирования временного ряда				
3.3	TSNA	4			
3.5	Адаптивные методы прогнозирования	4			
3.6	МР Временные ряды	2			

КИМ 10 класс

Примерная контрольная работа по дисциплине «**Предобработка данных**»

Примечание: Подразумевается, что все задания выполняются в среде Jupyter Notebook или в скрипте Python с использованием общепринятых библиотек. Ответом на многие задания является написанный код

Часть 1. Теоретические вопросы (кратко)

1. Математические основы Python (NumPy)

- В чем ключевое отличие массива `numpy.ndarray` от списка `list`? Приведите два преимущества NumPy с точки зрения производительности и функциональности.

2. Структурированные данные

- Какой формат структурированных данных (JSON, XML, CSV) вы бы выбрали для хранения иерархических данных (например, конфигурации программы) и почему? Для чего лучше подходит CSV?

3. Предобработка данных (Data Preprocessing)

- Перечислите 3 типичные проблемы в сырых данных и методы их обработки (например, способы обработки пропущенных значений).

Часть 2. Практические задания (код и объяснения)

Задание 1: Работа с табличными данными (Pandas)

Дан DataFrame `df`, загруженный из CSV-файла 'sales.csv' с

колонками: `Date`, `Product`, `Region`, `Sales`, `Price`.

```
python
import pandas as pd
df = pd.read_csv('sales.csv')
```

Напишите код, который:

1. Выведет первые 5 строк и общую информацию о DataFrame.
2. Покажет средние продажи (`Sales`) по каждому региону (`Region`).
3. Добавит новую колонку `Revenue`, которая равна `Sales * Price`.
4. Найдет продукт (`Product`) с максимальной выручкой (`Revenue`) за весь период.

Задание 2: Обработка текстовой информации

Дан список отзывов:

```
reviews = ["Этот товар просто отличный!", "Очень плохое качество.", "Нормально, но могло быть и лучше."]
```

Напишите код, который:

1. Приведет все тексты к нижнему регистру и удалит знаки препинания (хотя бы точки и восклицательные знаки).
2. Создаст новый список, содержащий только те отзывы, в которых больше 3 слов.

Задание 3: Работа с JSON

Дан JSON-объект в переменной `json_data`:

```
json
{
  "company": "TechCorp",
  "employees": [
    {"name": "Alice", "age": 30, "skills": ["Python", "SQL"]},
    {"name": "Bob", "age": 25, "skills": ["Java", "C++"]}
  ]
}
```

Напишите код, который:

1. Распарсит эту JSON-строку в словарь Python.
2. Выведет имя второго сотрудника в списке.
3. Добавит нового сотрудника {"name": "Charlie", "age": 35} в список employees и преобразует словарь обратно в JSON-строку.

Задание 4: Визуализация данных (Matplotlib)

Создайте DataFrame data со случайными данными:

```
python
import numpy as np
dates = pd.date_range(start='2023-01-01', periods=100, freq='D')
values = np.random.randn(100).cumsum() # случайные кумулятивные значения
data = pd.DataFrame({'Date': dates, 'Value': values})
data.set_index('Date', inplace=True)
```

Напишите код, который построит линейный график изменения Value по дате. Добавьте заголовок "Динамика показателя" и подписи к осям.

Задание 5: Предобработка данных

Дан DataFrame с пропусками и выбросами:

```
python
df_test = pd.DataFrame({
    'A': [1, 2, np.nan, 4, 5, 120], # выброс 120
    'B': ['a', 'b', 'b', np.nan, 'a', 'a'],
    'C': [10, 20, 30, 40, 50, 60]
})
```

Напишите код, который:

1. Заполнит пропуски в числовой колонке A её медианным значением, а в категориальной колонке B — самым частым значением (модой).
2. Обнаружит и заменит выбросы в колонке A (значения за пределами 3-х стандартных отклонений от среднего) на медианное значение этой колонки.

Часть 3. Задача на комплексное применение навыков

Задача: Анализ набора данных

Вам предоставлен набор данных о пассажирах Титаника в

файле `titanic.csv` (колонки: `Pclass`, `Name`, `Sex`, `Age`, `SibSp`, `Parch`, `Fare`, `Embarked`, `Survived`).

Напишите код, который выполняет следующий анализ:

- 1. Загрузка и первичный осмотр:** Загрузите данные, выведите количество строк и колонок, типы данных, проверьте наличие пропущенных значений.
- 2. Предобработка:**
 - Заполните пропуски в возрасте (`Age`) медианным возрастом.
 - Преобразуйте категориальный признак `Sex` в числовой (например, `male -> 0`, `female -> 1`).
 - Создайте новый признак `FamilySize` как сумму `SibSp` + `Parch` + 1 (сам пассажир).
- 3. Визуализация и анализ:**
 - Постройте гистограмму распределения возрастов пассажиров.
 - С помощью `seaborn` постройте `barplot`, показывающий средний уровень выживаемости (`Survived`) в зависимости от пола (`Sex`).
 - Посчитайте корреляцию между признаками `Survived`, `Pclass`, `Age`, `Fare`, `FamilySize` и визуализируйте её в виде тепловой карты (`heatmap`) с помощью `seaborn.heatmap()`.
- 4. Вывод:** Сделайте один краткий вывод по результатам визуализации (например, "На графике видно, что выживаемость сильно зависела от пола...").

Контрольная работа по теме: "Введение в машинное обучение"

Примечание: Для практических задач подразумевается использование библиотек `pandas`, `numpy`, `scikit-learn`, `matplotlib`. Ответом на многие задания является написанный код и интерпретация результатов.

Часть 1: Теоретические вопросы

1. Введение в машинное обучение

- Дайте определение машинного обучения. Чем оно принципиально отличается от классического программирования? Назовите три основные типа задач ML и приведите по одному примеру для каждого.

2. Регрессионный анализ моделей

- В чём суть задачи регрессии? Опишите принцип работы линейной регрессии. Что характеризуют коэффициенты в уравнении линейной регрессии $y = w_0 + w_1 * x_1$? Что такое ошибка (loss function) в данном контексте?

3. Ансамбли на основе деревьев решений

- Объясните принцип построения ансамблевых методов Bagging и Boosting. Чем случайный лес (Random Forest) отличается от градиентного бустинга (Gradient Boosting)?

4. Метод опорных векторов (SVM)

- Какова основная идея метода опорных векторов для линейно разделимой выборки? Что такое "зазор" (margin) и почему SVM стремится его максимизировать?

5. Байесовские методы классификации

- Запишите формулу теоремы Байеса. На каком упрощающем предположении основана работа наивного байесовского классификатора? Почему это предположение называется "наивным"?

6. Алгоритмы кластеризации данных

- Опишите алгоритм K-Means. Что является результатом его работы? Какой метрикой можно оценить качество кластеризации, если истинные метки кластеров неизвестны?

Часть 2: Практические задания

Задание 1: Регрессия и деревья решений

1. **Создайте и обучите модель линейной регрессии** на synthetic data.

python

```

from sklearn.datasets import make_regression
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

# Генерируем данные
X, y = make_regression(n_samples=100, n_features=1, noise=10, random_state=42)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Создайте и обучите модель линейной регрессии
lr = LinearRegression()
# ... (обучение и предсказание)
# Выведите коэффициент w1 и свободный член w0
# Оцените качество на тесте с помощью MSE

```

2. **Создайте и обучите модель решающего дерева** на этих же данных. Сравните MSE двух моделей. Какая из них показала лучший результат на этих данных? Почему?

Задание 2: Классификация (SVM и Наивный Байес)

```

python
from sklearn.datasets import load_iris
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Загружаем данные
iris = load_iris()
X, y = iris.data, iris.target
# Упростим задачу до бинарной классификации (классы 0 и 1)
X = X[y != 2]
y = y[y != 2]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

```

1. **Обучите классификатор опорных векторов (SVC)** с линейным ядром (`kernel='linear'`). Вычислите accuracy на тестовой выборке.
2. **Обучите наивный байесовский классификатор (GaussianNB)**. Вычислите accuracy на тестовой выборке.
3. *Вопрос:* Какая модель справилась лучше? Как вы думаете, почему?

Задание 3: Кластеризация (K-Means)

```
python
```

```
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Генерируем данные
X, y_true = make_blobs(n_samples=300, centers=4, cluster_std=0.60, random_state=0)
```

1. **Визуализируйте** сгенерированные данные с помощью scatter-plot.
2. **Обучите алгоритм K-Means** с `n_clusters=4`. Предскажите кластеры для данных.
3. **Визуализируйте результат кластеризации** (раскрасьте точки в соответствии с предсказанными кластерами, отметьте центроиды).
4. *Вопрос:* Совпали ли результаты кластеризации с истинным распределением? Как можно количественно оценить качество кластеризации для этого набора данных? (Назовите метрику).

Часть 3: Комплексная задача на предобработку и построение пайплайна

Задача: Предсказание стоимости домов

Вам дан dataset `house_prices.csv` (воображаемый) с признаками:

`Size, Bedrooms, Age, District, Price.`

Напишите код, который:

1. Загружает данные, проверяет на пропуски и выбросы.
2. Разделяет признаки и целевую переменную (`Price`). Разбивает данные на train/test.
3. Создает **пайплайн** (`sklearn.pipeline.Pipeline`), который:
 - Обрабатывает числовые признаки (например, стандартизирует `Size, Age`).
 - Обрабатывает категориальные признаки (например, one-hot кодирует `District`).
 - Применяет **ансамблирующую модель** на основе деревьев (например, `RandomForestRegressor` или `GradientBoostingRegressor`).
4. Обучает пайплайн на тренировочных данных.
5. Оценивает качество предсказания на тестовых данных с помощью **Mean Absolute Error (MAE)** и **R2-score**.
6. *Вопрос:* Какую важность имеют различные признаки в предсказании цены? Выведите и интерпретируйте `feature_importances_` вашей модели.

Контрольная работа по теме: "Продвинутые методы машинного обучения"

Примечание: Для выполнения заданий разрешается использовать `sklearn`, `numpy`, `pandas`, `matplotlib`, `scipy`. Ответом является код и краткое пояснение.

Часть 1: Теоретические вопросы (кратко)

1. Бэггинг-ансамбли моделей

- Объясните принцип бэггинга. Почему бэггинг эффективен для уменьшения дисперсии ошибки модели? Как получается итоговое предсказание в ансамбле при регрессии и классификации?

2. Ансамбли: Бустинг

- В чём ключевое отличие бустинга от бэггинга? Опишите алгоритм работы градиентного бустинга: как строится каждое последующее дерево и как учитываются ошибки предыдущих?

3. Ансамбли: Стекинг

- Что такое стекинг (stacking)? Опишите архитектуру стекинга: что такое базовые модели и мета-модель? Как избежать переобучения при обучении стекинга?

4. Матричные разложения: PCA и ICA

- Сформулируйте цель метода главных компонент (PCA). Что такое объяснённая дисперсия? Чем цель ICA принципиально отличается от цели PCA?

5. Обучение метрических преобразований

- Для чего используется метод метрического обучения (metric learning)? Приведите пример задачи, где он может быть полезен.

6. Подбор гиперпараметров

- Сравните методы поиска по сетке (GridSearch) и случайного поиска (RandomSearch). В каком случае RandomSearch может быть предпочтительнее?

7. Методы оптимизации (в ML)

- Опишите принцип работы градиентного спуска. Что такое скорость обучения (learning rate) и как она влияет на процесс оптимизации?

Часть 2: Практические задания

Задание 1: Сравнение бэггинга и бустинга

```
python
from sklearn.datasets import make_classification
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.model_selection import cross_val_score
import numpy as np

# Генерируем сложный dataset
X, y = make_classification(n_samples=1000, n_features=20, n_informative=15,
                           n_redundant=5, n_clusters_per_class=2, random_state=42)
```

1. **Создайте и протестируйте модель бэггинга** (RandomForestClassifier со 100 деревьями). Оцените её accuracy с помощью кросс-валидации (5 folds).
2. **Создайте и протестируйте модель бустинга** (GradientBoostingClassifier со 100 деревьями). Оцените accuracy.
3. **Сравните результаты** двух моделей по среднему accuracy и его стандартному отклонению. Какая модель показала себя лучше на этих данных?
4. **Постройте график важности признаков** для обеих моделей. Совпадают ли самые важные признаки в RandomForest и GradientBoosting?

Задание 2: Понижение размерности (PCA)

```
python
from sklearn.datasets import load_breast_cancer
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

data = load_breast_cancer()
X = data.data
y = data.target
```


1. **Стандартизируйте данные.**
2. **Примените PCA.** Постройте график объяснённой дисперсии в зависимости от числа компонент. Сколько компонент нужно, чтобы объяснить 95% дисперсии?
3. **Преобразуйте данные** в пространство главных компонент (выберите 2 компоненты). Визуализируйте данные в 2D, раскрасив точки согласно целевому признаку y . Можно ли увидеть разделение классов?

Задание 3: Подбор гиперпараметров (Optuna)

```
python
# Задание: Подберите гиперпараметры для GradientBoostingClassifier
# с помощью библиотеки Optuna (или GridSearchCV, если Optuna недоступна).
# Целевая метрика: accuracy.
# Параметры для подбора:
# - n_estimators (от 50 до 200)
# - learning_rate (лог-равномерно от 0.01 до 0.3)
# - max_depth (от 3 до 7)

import optuna
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Разделите данные на train и test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

def objective(trial):
    # Предложите параметры для испытания
    n_estimators = trial.suggest_int('n_estimators', 50, 200)
    learning_rate = trial.suggest_float('learning_rate', 0.01, 0.3, log=True)
    max_depth = trial.suggest_int('max_depth', 3, 7)

    # Создайте и обучите модель
    model = GradientBoostingClassifier(n_estimators=n_estimators,
                                       learning_rate=learning_rate,
                                       max_depth=max_depth,
                                       random_state=42)

    model.fit(X_train, y_train)
    # Оцените модель
    y_pred = model.predict(X_test)
    return accuracy_score(y_test, y_pred)

# Запустите изучение
study = optuna...
```

```
# Выведите лучшие параметры и accuracy
print(study.best_params)
print(study.best_value)
```

Задание 4: Построение стекинга (Stacking)

```
python
from sklearn.ensemble import StackingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier

# Определите базовые модели
base_models = [
    ('svm', SVC(probability=True, random_state=42)),
    ('dt', DecisionTreeClassifier(max_depth=5, random_state=42))
]

# Определите мета-модель
meta_model = LogisticRegression()

# Создайте стекинг-классификатор
stacking_clf = StackingClassifier

# Оцените его производительность с помощью кросс-валидации и сравните с каждой базово
й моделью в отдельности
# Вывод: Показывает ли стекинг улучшение по сравнению с лучшей базовой моделью?
```

Часть 3: Комплексная задача

Задача: Построение оптимизированного ансамбля

Используйте датасет `digits` (распознавание цифр).

1. **Предобработка:** Примените PCA для понижения размерности. Оставьте такое количество компонент, чтобы сохранить 0.99 дисперсии.
2. **Базовые модели:** Обучите 3 разнородные базовые модели (например, k-NN, SVM, Random Forest) на преобразованных данных.
3. **Ансамбль:** Постройте ансамбль методом **стекинга**. В качестве мета-модели используйте логистическую регрессию.

4. **Оптимизация:** С помощью **Optuna** подберите гиперпараметры для одной из самых важных базовых моделей (на ваш выбор) и для мета-модели.
5. **Сравнение:** Сравните ассигасу стекинга и лучшей базовой модели на тестовом наборе. Сделайте вывод об эффективности ансамбля.

Контрольная работа по теме: "Введение в нейронные сети"

Примечание: Для выполнения заданий разрешается использовать `torch`, `torchvision`, `numpy`, `matplotlib`. Ответом является код и краткое пояснение.

Часть 1: Теоретические вопросы

1. Полносвязные нейронные сети

- Опишите архитектуру многослойного перцептрона (MLP). Что такое функция активации и зачем она нужна? Приведите примеры трех нелинейных функций активации.

2. Обучение нейронных сетей

- Опишите алгоритм обратного распространения ошибки (backpropagation). Какую роль в этом процессе играет градиентный спуск? Дайте определение эпохи и итерации.

3. Введение в PyTorch

- Что такое `torch.Tensor` и чем он отличается от `numpy.ndarray`? Опишите назначение основных модулей PyTorch: `torch.nn`, `torch.optim`, `torch.utils.data`.

4. Ускорение обучения и снижение переобучения

- Что такое Dropout и Batch Normalization? Какую проблему решает каждый из этих методов? Как метод Dropout работает во время обучения и во время инференса?

5. Основы компьютерного зрения

- Чем полносвязные слои неудобны для обработки изображений? Сформулируйте ключевые идеи сверточных нейронных сетей (CNN): локальные связи, разделяемые веса, пулинг.

6. Методы оптимизации (в контексте НС)

- Сравните классический стохастический градиентный спуск (SGD) с оптимизатором Adam. В чем преимущества Adam? Что такое "learning rate scheduling" и зачем он нужен?

Часть 2: Практические задания на PyTorch

Задание 1: Создание и обучение простой нейронной сети

1. **Создайте модель:** Реализуйте на PyTorch полносвязную нейронную сеть (MLP) для классификации изображений из датасета FashionMNIST.
 - Архитектура: входной слой ($28 \times 28 = 784$ нейрона), один скрытый полносвязный слой (128 нейронов) с функцией активации ReLU, выходной слой (10 нейронов, по числу классов).
 - На выходе используйте `log_softmax`.

python

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.fc1 = nn.Linear(28 * 28, 128)
        self.fc2 = nn.Linear(128, 10)

    def forward(self, x):
        x = x.view(-1, 28 * 28) # Flatten the image
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return F.log_softmax(x, dim=1)
```

2. **Напишите цикл обучения:** Инициализируйте модель, функцию потерь (`nn.NLLLoss`), оптимизатор (`optim.Adam`). Напишите цикл на 5 эпох, который на каждой итерации:
 - обнуляет градиенты (`optimizer.zero_grad()`)

- делает forward pass
- вычисляет loss
- делает backward pass (`loss.backward()`)
- обновляет веса (`optimizer.step()`)
- выводит loss каждые 200 батчей.

Задание 2: Борьба с переобучением

Модифицируйте модель из Задания 1:

1. **Добавьте Dropout** (с вероятностью 0.25) после активации скрытого слоя.
2. **Добавьте Batch Normalization** после скрытого слоя, но до функции активации.
3. **Обучите новую модель.** Сравните графики loss на тренировочном и валидационном наборах до и после добавления этих слоев. Сделайте вывод об эффекте.

Задание 3: Переход к компьютерному зрению (CNN)

1. **Создайте сверточную нейронную сеть (CNN)** для той же задачи.
 - Архитектура: Conv2d(1, 32, 3) -> ReLU -> MaxPool2d(2) -> Conv2d(32, 64, 3) -> ReLU -> MaxPool2d(2) -> Flatten -> Linear(64 * 5 * 5, 128) -> ReLU -> Linear(128, 10).
 - *Вопрос:* Почему на входе линейного слоя стоит 64 * 5 * 5? Рассчитайте это значение.

python

```
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, 3)
        self.conv2 = nn.Conv2d(32, 64, 3)
        self.fc1 = nn.Linear(64 * 5 * 5, 128) # !!! Рассчитайте размерность
        self.fc2 = nn.Linear(128, 10)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = F.max_pool2d(x, 2)
        x = F.relu(self.conv2(x))
        x = F.max_pool2d(x, 2)
        x = x.view(-1, 64 * 5 * 5) # Flatten
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return F.log_softmax(x, dim=1)
```

2. **Обучите CNN** на 5 эпохах.

3. **Сравните ассурасу** на тестовом наборе у простой MLP и у CNN. Какая архитектура справилась лучше и почему?

Задание 4: Использование готовых датасетов и DataLoader

1. **Загрузите данные** FashionMNIST с помощью `torchvision.datasets.FashionMNIST` и преобразуйте их в тензоры.
2. **Создайте DataLoader** для тренировочного и тестового наборов с `batch_size=64` и `shuffle=True` для тренировочного.
3. **Модифицируйте цикл обучения** из Задания 1, чтобы он итерировался по батчам из `DataLoader`.

Часть 3: Комплексная задача

Задача: Эксперимент с оптимизацией

Цель: Исследовать влияние различных техник на процесс обучения.

1. **Базовый уровень:** Обучите CNN из Задания 3 на 10 эпохах. Зафиксируйте итоговую ассурасу на валидационной выборке.
2. **Эксперименты:** Поочередно измените следующие параметры и повторите обучение:
 - **Оптимизатор:** Замените Adam на SGD.
 - **Schedule:** Добавьте `torch.optim.lr_scheduler.StepLR`, который уменьшает learning rate в 10 раз каждые 5 эпох.
 - **Аугментация:** Добавьте аугментацию данных для тренировочного набора (например, `torchvision.transforms.RandomHorizontalFlip()`).
3. **Анализ:** Для каждого эксперимента зафиксируйте ассурасу. Какой метод дал наибольший прирост в качестве? Постройте графики сходимости (loss по эпохам) для базового уровня и для самого удачного эксперимента. Сделайте вывод.

Контрольная работа по теме: "Продвинутые методы компьютерного зрения"

Часть 1: Теоретические вопросы

1. Виды сверток

- Объясните принцип работы depthwise separable convolution. В чем ее преимущество перед обычной сверткой?
- Что такое dilated convolution и для каких задач она применяется?

2. Методы оптимизации

- Сравните Adam и SGD с моментом. В каких случаях каждый из них предпочтительнее?
- Что такое learning rate scheduling? Приведите примеры методов планирования.

3. Регуляризация

- Объясните принцип работы Dropout. Как он помогает бороться с переобучением?
- Что такое Batch Normalization и как она ускоряет обучение?

4. Аугментация данных

- Какие методы аугментации эффективны для задач детекции объектов?
- Опишите принцип работы CutMix и MixAugment.

5. Перенос обучения

- Что такое feature extraction и fine-tuning? Когда какой подход использовать?
- Как выбрать learning rate для fine-tuning предобученной модели?

6. Детекция объектов

- Опишите архитектуру YOLOv5. В чем ее преимущества перед предыдущими версиями?
- Что такое anchor boxes и как они настраиваются?

7. Семантическая сегментация

- Чем отличается семантическая сегментация от instance segmentation?
- Опишите принцип работы U-Net архитектуры.

Часть 2: Практические задания

Задание 1: Реализация различных сверток

```
python
import torch
import torch.nn as nn

# 1. Реализуйте depthwise separable convolution
class DepthwiseSeparableConv(nn.Module):
    def __init__(self, in_channels, out_channels, kernel_size):
        super().__init__()
        self.depthwise = nn.Conv2d(_____, _____, _____,
                                     groups=_____, padding='same')
        self.pointwise = nn.Conv2d(_____, _____, _____)

    def forward(self, x):
        return self.pointwise(_____)

# 2. Создайте dilated convolution с коэффициентом 2
dilated_conv = nn.Conv2d(3, 16, _____, _____=2, _____=2)
```

Задание 2: Настройка оптимизации

```
python
from torch.optim import _____, _____
from torch.optim.lr_scheduler import _____

# 1. Создайте оптимизатор SGD с моментом и weight decay
optimizer = _____(model.parameters(), lr=0.1, _____=0.9, _____=1e-4)

# 2. Добавьте Cosine Annealing scheduler
scheduler = _____(optimizer, T_max=100)

# 3. Реализуйте warm-up
def warmup_lr_scheduler(optimizer, warmup_iters, warmup_factor):
    def f(x):
        if x >= warmup_iters:
            return 1
        alpha = float(x) / warmup_iters
        return warmup_factor * (1 - alpha) + alpha
    return torch.optim.lr_scheduler.LambdaLR(optimizer, f)
```

Задание 3: Аугментация для детекции

```
python
```



```

import albumentations as A
from albumentations.pytorch import ToTensorV2

transform = A.Compose([
    A.HorizontalFlip(_____=0.5),
    A.RandomBrightnessContrast(_____=0.2),
    A.ShiftScaleRotate(_____=0.1, _____=0.2, _____=15, p=0.5),
    A.Cutout(_____=8, max_h_size=16, max_w_size=16, p=0.5),
    A.Normalize(mean=(0.485, 0.456, 0.406), std=(0.229, 0.224, 0.225)),
    _____
], bbox_params=A.BboxParams(format='_____', min_visibility=0.3))

```

Задание 4: Fine-tuning предобученной модели

```

python
from torchvision import models

# 1. Загрузите предобученную ResNet50
model = models._____(pretrained=_____)

# 2. Заморозьте все слои кроме последних двух
for _____ in model.parameters():
    _____.requires_grad = _____

for param in model._____.parameters():
    param.requires_grad = _____

# 3. Замените классификатор
num_features = model._____.in_features
model.fc = nn.Sequential(
    nn._____(0.3),
    nn._____(num_features, 256),
    nn._____( ),
    nn._____(256, 10)
)

```

Задание 5: Детекция объектов YOLO

```

python
from ultralytics import YOLO

# 1. Загрузка предобученной модели
model = YOLO('_____.pt')

# 2. Конфигурация обучения
results = model.train(
    data='coco8.yaml',
    epochs=_____,
    imgsz=_____,

```

```

        batch=_____,
        optimizer='_____',
        lr0=0.01,
        _____=0.9,
        _____=0.0005
    )

# 3. Инференс
results = model('image.jpg')
results[0]._____()

```

Задание 6: Семантическая сегментация

```

python
import segmentation_models_pytorch as _____

# 1. Создайте U-Net модель
model = _____.Unet(
    encoder_name='_____',
    encoder_weights='_____',
    classes=_____
)

# 2. Реализуйте Dice Loss
class DiceLoss(nn.Module):
    def __init__(self):
        super().__init__()

    def forward(self, pred, target):
        smooth = 1e-6
        pred = pred._____()
        intersection = (_____ * _____).sum(dim=(2,3))
        union = pred.sum(dim=(2,3)) + target.sum(dim=(2,3))
        dice = (_____ * intersection + smooth) / (_____ + smooth)
        return 1 - dice._____()

# 3. Реализуйте метрику IoU
def calculate_iou(pred_mask, true_mask):
    intersection = (_____ & _____).float().sum((1, 2))
    union = (_____ | _____).float().sum((1, 2))
    iou = (_____ + 1e-6) / (_____ + 1e-6)
    return iou._____()

```

Часть 3: Комплексные задачи

Задача 1: Сравнение архитектур

Сравните эффективность различных сверточных блоков:

1. Обычная свертка
2. Depthwise separable convolution
3. Dilated convolution

Для сравнения используйте:

- Количество параметров
- FLOPs
- Точность на validation set
- Время обучения

Задача 2: Оптимизация гиперпараметров

Подберите оптимальные гиперпараметры для YOLOv5:

- Learning rate
- Optimizer (SGD/Adam)
- Weight decay
- Augmentation policy

Обоснуйте выбор метрик для оценки качества.

Задача 3: Полный пайплайн

Разработайте полный пайплайн для задачи семантической сегментации:

1. Предобработка данных
2. Аугментация
3. Модель (U-Net)
4. Функция потерь
5. Метрики качества
6. Оптимизация

Критерии оценки:

- Правильность заполнения пропусков в коде
- Глубина понимания теоретических концепций
- Качество анализа в комплексных задачах
- Обоснованность предлагаемых решений
- Оптимальность выбранных параметров

Контрольная работа по теме: "Основы обработки естественного языка"

Часть 1: Теоретические вопросы

1. Векторные представления текста

- Дайте определение эмбедингов (word embeddings). Объясните, чем они принципиально отличаются от one-hot кодирования.
- Опишите разницу между подходами Bag of Words (BoW) и TF-IDF. В каких задачах какой подход предпочтительнее?

2. Предобработка текстовых данных

- Перечислите основные этапы предобработки текста. Почему лемматизация часто дает лучшие результаты, чем стемминг?
- Что такое токенизация? Объясните разницу между word-level и subword-level токенизацией.

3. Word2Vec

- Опишите архитектуры Skip-gram и CBOW. Какая из них лучше работает с редкими словами и почему?
- Объясните принцип работы negative sampling в Word2Vec.

4. Рекуррентные нейронные сети

- В чем заключается проблема исчезающих градиентов в обычных RNN?
- Опишите архитектуру LSTM. Как работают forget gate и input gate?

5. Метрики оценки в NLP

- Какие метрики используются для оценки качества: а) классификации текста б) машинного перевода?

Часть 2: Практические задания

Задание 1: Предобработка текста

Напишите функцию на Python, которая:

1. Приводит текст к нижнему регистру
2. Удаляет пунктуацию и цифры
3. Проводит токенизацию
4. Удаляет стоп-слова
5. Выполняет лемматизацию

Задание 2: Создание матрицы TF-IDF

Дан корпус из трех документов:

1. "the cat sat on the mat"
2. "the dog sat on the log"
3. "cats and dogs are animals"

Рассчитайте вручную или напишите код для создания TF-IDF матрицы.

Задание 3: Word2Vec

Напишите код для обучения Word2Vec модели на трех предложениях:

- ["cat", "sat", "on", "mat"]
- ["dog", "sat", "on", "log"]
- ["cats", "and", "dogs", "are", "pets"]

Параметры: размерность векторов - 100, window size - 2, алгоритм - skip-gram.

Задание 4: Построение RNN

Создайте класс простой RNN для классификации текста на PyTorch со следующими слоями:

- Embedding слой
- RNN слой
- Полносвязный слой

Опишите процесс обучения такой модели.

Задание 5: Реализация LSTM

Модифицируйте RNN из предыдущего задания в LSTM. Объясните, какие изменения нужно внести и почему они улучшат работу модели.

Часть 3: Комплексные задачи

Задача 1: Сравнение подходов

Сравните три подхода к векторному представлению текста:

1. TF-IDF
2. Word2Vec
3. Embedding слой в нейронной сети

Для каждого подхода укажите:

- Преимущества и недостатки
- Области применения
- Вычислительную сложность

Задача 2: Анализ архитектур

Проанализируйте эволюцию архитектур для работы с текстом:

- От простых RNN к LSTM/GRU
- От рекуррентных сетей к трансформерам

Объясните, какие проблемы решает каждая последующая архитектура.

Задача 3: Проектирование системы

Разработайте архитектуру системы для классификации тональности отзывов. Система должна включать:

- Предобработку текста
- Векторизацию
- Модель машинного обучения
- Оценку качества

Обоснуйте выбор каждого компонента.

Контрольная работа по теме: "Анализ и прогнозирование временных рядов"

Часть 1: Теоретические вопросы

1. Анализ временных рядов

- Какие компоненты временного ряда вы знаете? Опишите каждую из них.
- Что такое стационарность временного ряда? Почему это важно для прогнозирования?

2. Машинное обучение для прогнозирования

- Какие особенности временных рядов важно учитывать при построении ML-моделей?
- Чем подходы к прогнозированию временных рядов отличаются от обычных задач ML?

3. TSA (Time Series Analysis)

- Что такое автокорреляция (ACF) и частичная автокорреляция (PACF)? Как они используются при подборе параметров ARIMA-моделей?
- Опишите процедуру проверки временного ряда на стационарность.

4. Адаптивные методы прогнозирования

- В чем преимущество адаптивных методов перед статическими?
- Опишите принцип работы экспоненциального сглаживания.

5. Кросс-валидация для временных рядов

- Почему для временных рядов нельзя использовать случайную кросс-валидацию?
- Опишите принцип работы кросс-валидации с расширяющимся окном.

Часть 2: Практические задания

Задание 1: Анализ стационарности

```
python
import pandas as pd
import numpy as np
from statsmodels.tsa.stattools import adfuller
import matplotlib.pyplot as plt

# Загрузите временной ряд
data = pd.read_csv('time_series.csv', index_col='date', parse_dates=True)

# 1. Постройте график временного ряда
plt.figure(figsize=(12, 6))
plt.plot(_____)
plt.title('Временной ряд')
plt.show()

# 2. Проверка на стационарность с помощью теста Дики-Фуллера
result = adfuller(_____)
print('ADF Statistic:', result[0])
print('p-value:', result[1])
print('Critical Values:')
for key, value in result[4].items():
    print(f' {key}: {value}')

# 3. Если ряд нестационарный, примените дифференцирование
data['diff'] = data['value']._____(periods=1)
```

Задание 2: Построение ARIMA модели

```
python
```



```

from statsmodels.tsa.arima.model import ARIMA
from sklearn.metrics import mean_absolute_error, mean_squared_error

# 1. Разделение на train/test (последние 30 точек в test)
train = data[:-30]
test = data[-30:]

# 2. Подбор параметров ARIMA(p,d,q)
model = ARIMA(_____, order=(1, 1, 1))
model_fit = model._____( )

# 3. Прогноз на test
forecast = model_fit._____(start=len(train), end=len(train)+len(test)-1)

# 4. Оценка качества
mae = mean_absolute_error(_____, _____)
rmse = np.sqrt(mean_squared_error(_____, _____))
print(f'MAE: {mae:.2f}, RMSE: {rmse:.2f}')

```

Задание 3: Экспоненциальное сглаживание

```

python
from statsmodels.tsa.holtwinters import ExponentialSmoothing

# 1. Построение модели с трендом и сезонностью
model = ExponentialSmoothing(_____,
                              trend='add',
                              seasonal='add',
                              seasonal_periods=12)
model_fit = model._____( )

# 2. Прогноз на 10 шагов вперед
forecast = model_fit._____(steps=10)

# 3. Визуализация результатов
plt.figure(figsize=(12, 6))
plt.plot(_____, label='Исторические данные')
plt.plot(_____, label='Прогноз')
plt.legend()
plt.show()

```

Задание 4: Машинное обучение для временных рядов

```

python
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler

# 1. Создание признаков (feature engineering)
def create_features(df):

```

```

df['hour'] = df.index._____
df['dayofweek'] = df.index._____
df['quarter'] = df.index._____
df['month'] = df.index._____
df['year'] = df.index._____
df['dayofyear'] = df.index._____
return df

data = create_features(data)

# 2. Создание лаговых признаков
for lag in range(1, 4):
    data[f'lag_{lag}'] = data['value']._____(lag)

# 3. Подготовка данных для ML
features = ['hour', 'dayofweek', 'quarter', 'month', 'year', 'dayofyear', 'lag_1', 'lag_2', 'lag_3']
X = data[features]
y = data['value']

# 4. Разделение на train/test
X_train, X_test = X[:-30], X[-30:]
y_train, y_test = y[:-30], y[-30:]

# 5. Обучение модели
model = _____(n_estimators=100, random_state=42)
model._____(X_train, y_train)

# 6. Прогноз и оценка
predictions = model._____(X_test)
mae = mean_absolute_error(_____, _____)

```

Задание 5: Кросс-валидация для временных рядов

```

python
from sklearn.model_selection import TimeSeriesSplit

tscv = TimeSeriesSplit(_____=5)

for train_index, test_index in tscv._____(X):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

    # Обучение и оценка модели на каждом fold
    model = RandomForestRegressor(n_estimators=100)
    model.fit(_____, _____)
    score = model.score(_____, _____)
    print(f'R² score: {score:.3f}')

```

Часть 3: Комплексные задачи

Задача 1: Сравнение методов прогнозирования

Сравните следующие методы на реальном временном ряде:

1. ARIMA
2. Экспоненциальное сглаживание
3. Случайный лес
4. XGBoost

Для сравнения используйте метрики:

- MAE (Mean Absolute Error)
- RMSE (Root Mean Square Error)
- MAPE (Mean Absolute Percentage Error)

Задача 2: Анализ остатков

Для лучшей модели из Задачи 1:

1. Постройте график остатков
2. Проверьте остатки на автокорреляцию (тест Льюнга-Бокса)
3. Проверьте нормальность распределения остатков
4. Сделайте вывод о качестве модели

Задача 3: Построение прогнозной системы

Разработайте архитектуру системы прогнозирования, которая включает:

1. Предобработку данных
2. Генерацию признаков
3. Выбор и обучение модели
4. Оценку качества
5. Обновление прогнозов

Опишите, как система *будет обрабатывать новые данные.*