Государственное бюджетное учреждение дополнительного образования
Дворец детского (юношеского) творчества Петродворцового района Санкт-Петербурга
«ПЕТЕРГОФ»

ПРИНЯТА
Педагогическим советом
ДДЮТ «ПЕТЕРГОФ»
Протокол № 1 от «31» августа 2026 г.

УТВЕРЖДЕНА
Приказом № 392/у от «01» сентября 2026 г.
Директор ДДЮТ «ПЕТЕРГОФ»
_____ В.А. Апраксимов

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 00C9DF420E45C5C12E918B262C7A3C1870
Владелец: Апраксимов Владислав Анатольевич
Действителен: с 14.04.2026 до 08.07.2027

Дополнительная общеразвивающая программа

«МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ БОЛЬШИХ ДАННЫХ»

Срок освоения: 1 год
Возраст обучающихся: 14-17 лет

Разработчик
Найк Полина Леонидовна
Педагог дополнительного образования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Настоящая программа составлена с учетом требований современных нормативных документов в области дополнительного образования, а также устава и соответствующих локальных актов ДДЮТ «ПЕТЕРГОФ».

Дополнительная общеразвивающая программа «Машинное обучение и анализ больших данных» (далее – образовательная программа) является одним из актуальных и востребованных способов изучения современных информационных технологий и искусственного интеллекта. Занятия по программе позволят учащимся развить алгоритмическое и логическое мышление, аналитические способности, навыки работы с большими объёмами данных, творческое воображение.

Учащиеся, осваивая навыки программирования на языке Python и методы машинного обучения, создают собственные проекты по анализу данных, построению предсказательных моделей, кластеризации и классификации информации. От простых линейных регрессий до ансамблевых методов (случайный лес) – программа выстраивает постепенный путь от новичка до уверенного пользователя инструментов Data Science.

Современные облачные и локальные среды разработки позволяют учащимся быстро приобщиться к миру анализа больших данных, превращая обучение в увлекательное исследование. В результате выполнения последовательных практических задач складывается полноценный проект, в котором применяются методы машинного обучения для решения реальных или учебных задач. Эта новая технологическая среда позволяет ребятам в полной мере раскрыть свои творческие и исследовательские способности. Учащиеся постепенно и в доступной форме научатся основам алгоритмизации на Python, познакомятся с библиотеками Pandas, NumPy, Scikit-learn, а также с технологией построения пайплайнов обработки данных, что позволит им в дальнейшем более эффективно изучать продвинутые разделы искусственного интеллекта. Содержание программы дает возможность учащимся приобрести навыки не только в программировании, но и в таких областях как статистика, анализ данных и визуализация информации.

Направленность и уровень освоения

Дополнительная общеразвивающая программа «Машинное обучение и анализ больших данных» по **направленности является технической**, по **уровню освоения – общекультурной (с элементами углублённого изучения)**.

Адресат программы

Программа предназначена для учащихся в возрасте от 14 до 17 лет, независимо от пола, с разной степенью сформированности интересов и мотивации к изучению машинного обучения и анализа данных.

Для успешного освоения программы желательно наличие у учащегося базовой компьютерной грамотности, включающей уверенное владение компьютером, навыки работы с файлами и папками, а также умение пользоваться браузером. В области математики приветствуется владение арифметическими действиями, понимание процентов, пропорций и среднего арифметического, умение работать с координатами на плоскости. От учащегося ожидается способность описать последовательность действий для решения простой задачи и найти закономерность в наборе чисел или фигур.

Важны также интерес к решению нестандартных задач, готовность пробовать, ошибаться и самостоятельно искать способы исправить ошибку, а также базовое знание английского языка на уровне понимания простых слов, используемых в программировании.

Для зачисления на программу проводится входное тестирование, включающее письменную часть и устную беседу. Тестирование позволяет определить уровень подготовки

учащегося, выявить темы, требующие дополнительного внимания, и предложить индивидуальную траекторию обучения. Учащиеся, показавшие по результатам тестирования недостаточный уровень подготовки, могут быть зачислены на программу после собеседования с педагогом и при условии готовности к дополнительным занятиям для устранения пробелов.

Наличие углублённых знаний в области программирования не является обязательным, однако приветствуется. Допускается разновозрастной состав групп. Наполняемость групп составляет не менее 12 человек.

Актуальность

Образовательная программа ориентирована на реализацию интересов учащихся в сфере анализа данных, алгоритмизации, машинного обучения и искусственного интеллекта, развитие их информационной и технологической культуры. Программа направлена на формирование познавательной мотивации, приобретение опыта продуктивной творческой, исследовательской и технической деятельности в области Data Science.

Актуальность программы также обусловлена тем, что работа над проектами по обработке данных открывает обучающимся путь к исследовательской деятельности, развивает аналитическое и системное мышление и предоставляет новые возможности для интеллектуального развития. Предполагается развитие обучающегося в самых различных направлениях: конструкторское мышление, математическая логика, образное и пространственное мышление, умение работать с большими объёмами информации. Все это необходимо современному человеку, чтобы реализовать себя в самых разных областях жизни, в том числе в будущей профессии в области IT и Data Science.

При реализации программы осуществляется взаимодействие педагога с обучающимся в сотрудничестве, используются доступные для учащихся понятия и термины. Системность, последовательность и доступность излагаемого материала, изучение нового материала опирается на ранее приобретенные знания с преобладающим приоритетом практической деятельности (решение задач на Python, анализ реальных датасетов, обучение моделей). Развитие в учащихся самостоятельности, ответственности, творчества и изобретательности является одним из основных приоритетов данной программы.

Объем и срок освоения программы

Программа рассчитана на **1 год обучения**, всего – **144 часов**, занятия проводятся **2 раза в неделю по 2 академических часа**.

Цель и задачи

Цель: приобретение учащимися базовых знаний, умений и навыков в области машинного обучения и анализа данных, формирование устойчивого интереса к Data Science и программированию на Python.

Задачи:

обучающие:

- способствовать овладению навыками составления алгоритмов и написания кода на языке Python;
- сформировать навыки работы с библиотеками Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn;
- освоить методы машинного обучения: линейную и логистическую регрессию, деревья решений, случайный лес, кластеризацию (K-means);
- научить оценивать качество моделей с помощью метрик (accuracy, precision, recall, F1, R^2, MSE);

развивающие:

- развить умение работать с большими объёмами данных, компьютерными программами и дополнительными источниками информации;
- способствовать развитию логического, критического, системного, алгоритмического и креативного мышления, аналитических способностей;
- способствовать развитию внимания, памяти, наблюдательности, познавательного интереса при работе с данными;

воспитательные:

- способствовать формированию положительного отношения к информатике, программированию и искусственному интеллекту;
- формирование умения работать в паре, малой группе, коллективе при решении проектных задач;
- формирование умения демонстрировать результаты своей работы в виде презентаций и защит проектов.

Планируемые результаты

В результате освоения программы учащиеся должны достигнуть следующих результатов:

Предметные

Учащиеся будут знать/понимать:

- основы синтаксиса языка Python (переменные, циклы, условия, функции, списки, словари);
- назначение и использование основных библиотек: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn;
- сущность алгоритмов машинного обучения: регрессии, классификации, кластеризации;
- базовые метрики оценки качества моделей.

Учащиеся будут уметь/иметь:

- писать простые программы на Python для обработки данных;
- загружать, очищать, преобразовывать и визуализировать данные из CSV-файлов;
- обучать модели машинного обучения (линейную регрессию, логистическую регрессию, случайный лес);
- оценивать качество моделей с помощью метрик и выбирать наилучшую.

Метапредметные

У учащихся разовьются:

- способность самостоятельно искать, анализировать, отбирать необходимую информацию (в том числе в документации библиотек и на профессиональных форумах);
- внимание, память, наблюдательность при работе с данными (согласно возрасту);
- воображение, логическое мышление и аналитические способности, творческие способности, стремление к достижению поставленной цели (согласно возрасту);
- коммуникативная компетентность в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности (парное программирование, групповые проекты).

Личностные

Учащимся будут созданы условия для формирования:

- положительного отношения к информатике, программированию и машинному обучению;
- устойчивой учебной мотивации к занятиям Data Science;
- ответственности за свою работу (контроль дедлайнов, доведение проектов до конца).

Организационно-педагогический условия реализации программы

Язык реализации программы

Дополнительная общеразвивающая программа «Машинное обучение и анализ больших данных» реализуется на **русском языке** (с использованием англоязычной терминологии в коде и библиотеках).

Форма обучения

Программа реализуется **очно.**

Программа предусматривает возможность организации образовательного процесса в очно-заочном или заочном формате с применением электронного обучения и дистанционных образовательных технологий. В течение года в случае возникновения сложной эпидемиологической обстановки возможен переход полностью или частично на электронное обучение с применением дистанционных технологий с учетом требований действующих Санитарно-эпидемиологических правил.

Условия набора и формирования групп. На обучение принимаются все желающие учащиеся в возрасте от 14 до 17 лет при условии успешного прохождения входного тестирования и собеседования с педагогом. Группы формируются из учащихся, показавших сходный уровень подготовки по результатам входного тестирования, что позволяет выстроить оптимальную траекторию обучения для каждого состава. Допускается формирование как одновозрастных, так и разновозрастных групп.

Наполняемость групп составляет не менее 12 человек. В случае наличия свободных мест возможен добор учащихся в течение учебного года при условии успешного прохождения входного тестирования.

Особенности организации образовательного процесса

По структуре данная программа является линейно-концентрической: учебный материал представлен так, чтобы каждый новый модуль опирался на пройденный. Принцип организации: от простого к сложному, от кода к данным, от данных к моделям.

Формы организации занятий

Программой предусматривается проведение аудиторных занятий в учебном кабинете под непосредственным руководством педагога.

Формы проведения занятий: лекции, мастер-классы, практические занятия, хакатоны, соревнования внутри группы, защита проектов. Основным видом учебной деятельности является выполнение практических работ на компьютере.

Формы организации деятельности учащихся на занятии:

фронтальная (работа со всеми учащимися одновременно), групповая (работа в малых группах, в том числе в парах) и индивидуальная (для коррекции пробелов и выполнения продвинутых заданий).

Материально-техническое обеспечение программы

Немаловажную роль играет современное материально-техническое обеспечение

программы. Занятия проводятся в светлом, просторном помещении, соответствующим нормам СанПиН.

– компьютерные столы и стулья (по количеству учащихся);

– компьютеры или ноутбуки с операционной системой WINDOWS XP/7,10+ (по количеству учащихся);

– локальная сеть локальная сеть с доступом в Интернет на каждого учащегося;

– маркерная доска или интерактивная панель – 1 шт.;

– мультимедийный проектор или телевизор с возможностью подключения к компьютеру педагога – 1 шт.;

– источник бесперебойного питания для рабочих станций.

Программное обеспечение и средства обучения:

- операционная система на компьютерах учащихся и педагога (Linux с графической оболочкой или аналоги);
- браузер (Google Chrome, Яндекс.Браузер или аналог с поддержкой современных веб-стандартов);
- среда разработки на языке Python, работающая в формате интерактивных блокнотов (локальная установка Jupyter Notebook / JupyterLab или VS Code с соответствующими расширениями);
- интерпретатор Python версии 3.9 или выше;
- набор библиотек: pandas, numpy, matplotlib, seaborn, scikit-learn;
- текстовый редактор для работы с файлами .py и .ipynb;
- программное обеспечение для видеоконференцсвязи (для организации дистанционных занятий при необходимости).

Облачные ресурсы (при использовании соответствующего формата работы):

- доступ к российской облачной среде Yandex DataSphere или аналогичным платформам (с соблюдением требований Федерального закона № 152-ФЗ «О персональных данных» и при наличии письменных согласий родителей);
- доступ к образовательным онлайн-платформам для поиска датасетов под контролем педагога.

Учебно-методическое обеспечение:

- раздаточные материалы: бланки входного тестирования, карточки с заданиями для текущего контроля, чек-листы для самооценки;
- доступ к электронным образовательным ресурсам: авторским презентациям педагога, видеоурокам, интерактивным конспектам;
- методическая литература и справочные пособия для педагога и учащихся (в бумажном или электронном виде).

УЧЕБНЫЙ ПЛАН

Учебный план 1 года обучения

№ п/п	Название раздела, темы	Количество часов			Формы контроля/ аттестации
		Всего	Теория	Практика	
1.	Основы Python и Git	38	10	28	Практическая работа
2.	Pandas, EDA, визуализация	31	6	25	Практическая работа
3.	Регрессия и классификация	30	7,5	22,5	Практическая работа
4.	Деревья, лес, кластеризация	22	5	17	Практическая работа
5.	Финальный проект + защита	20	0	20	Защита проекта
6.	Контрольные и итоговые занятия	3	0	3	Практическая работа, защита проекта
Итого:		**144**	**28,5**	**115,5**	

РАБОЧАЯ ПРОГРАММА

Пояснительная записка

Рабочая программа составлена на основе дополнительной общеразвивающей программы «МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ БОЛЬШИХ ДАННЫХ», имеет техническую направленность, общекультурный уровень освоения.

Программа предназначена для учащихся в возрасте от 14 до 17 лет, независимо от пола, с разной степенью сформированности интересов и мотивации к изучению машинного обучения и анализа данных., 150 часов в год, разработана педагогом дополнительного образования ДДЮТ «ПЕТЕРГОФ» Найк Полиной Леонидовной.

1 год обучения

Условия реализации программы на 1 году обучения

Первый год обучения рассчитан на **150 часов** за учебный год, 4 часа в неделю: теоретические и практические занятия 2 раза в неделю по 2 академических часа.

Возраст учащихся: 14–17 лет. Количество детей в группе: не менее 12 человек.

Особенности 1 года обучения

На первом году обучения основное внимание уделяется обучению **программированию на Python** и основам **анализа данных и машинного обучения**. Учащиеся осваивают работу с библиотеками Pandas, NumPy, Scikit-learn, учатся визуализировать данные, строить и оценивать модели машинного обучения.

Программа выстроена по принципу «от простого к сложному»: от базовых конструкций Python к работе с данными, затем к простым моделям ML и, наконец, к ансамблевым методам и кластеризации. Завершается курс выполнением и защитой индивидуального или группового проекта.

Задачи 1 года обучения:

обучающие:

- способствовать овладению навыками составления алгоритмов и написания кода на языке Python;
- сформировать навыки работы с библиотеками Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn;
- освоить методы машинного обучения: линейную и логистическую регрессию, деревья решений, случайный лес, кластеризацию (K-means);
- научить оценивать качество моделей с помощью метрик (accuracy, precision, recall, F1, R^2, MSE).

развивающие:

- развить умение работать с большими объёмами данных и дополнительными источниками информации;
- способствовать развитию логического, критического, системного, алгоритмического и творческого мышления, аналитических способностей;
- способствовать развитию внимания, памяти, наблюдательности, познавательного интереса при работе с данными.

воспитательные:

- способствовать формированию положительного отношения к информатике, программированию и искусственному интеллекту;
- формирование умения работать в паре, малой группе, коллективе при решении проектных задач;
- формирование умения демонстрировать результаты своей работы в виде презентаций и защит проектов.

Планируемые результаты 1 года обучения

Предметные результаты

Учащиеся будут знать/понимать:	Учащиеся будут уметь/иметь:
– основы синтаксиса языка Python (переменные, циклы, условия, функции, списки, словари);. – назначение и использование основных библиотек: Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn; – сущность алгоритмов машинного обучения: регрессии, классификации, кластеризации; – базовые метрики оценки качества моделей.	– писать простые программы на Python для обработки данных; – загружать, очищать, преобразовывать и визуализировать данные из CSV-файлов; – обучать модели машинного обучения (линейную регрессию, логистическую регрессию, случайный лес); – оценивать качество моделей с помощью метрик и выбирать наилучшую.

Метапредметные результаты

У учащихся разовьются:

– способность самостоятельно искать, анализировать, отбирать необходимую информацию (в том числе в документации библиотек и на профессиональных форумах);

– внимание, память, наблюдательность, познавательный интерес (согласно возрасту);

– воображение, логическое мышление и аналитические способности, творческие способности, стремление к достиж поставленной цели (согласно возрасту);

– коммуникативная компетентность в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности (парное программирование, групповые проекты).

Личностные результаты.

Учащимся будут созданы условия для формирования:

– положительного отношения к информатике, программированию и машинному обучению;

– устойчивой учебной мотивации к занятиям Data Science;

– ответственности за свою работу (контроль дедлайнов, доведение проектов до конца).

СОДЕРЖАНИЕ ПРОГРАММЫ
1 год обучения

1. Основы Python и Git

1.1. Введение. Техника безопасности. Знакомство со средой разработки

Теория. Правила по технике безопасности в учреждении. Рассказ о развитии искусственного интеллекта и машинного обучения. Знакомство со средой разработки (Jupyter Notebook / Google Colab / Yandex DataSphere).

Практика. Первый print, арифметические операции (+, -, *, /, //, %), входная диагностика: опрос.

1.2. Переменные, типы данных, ввод/вывод

Теория. Понятие переменной. Правила именования переменных. Типы данных: int, float, str. Функции input() и print(). Форматированный вывод (f-строки). Преобразование типов (int(), str(), float()).

Практика. Решение задач: калькулятор, обмен значениями двух переменных, ввод данных пользователя с выводом приветствия.

1.3. Условный оператор if-elif-else

Теория. Операторы сравнения (==, !=, >, <, >=, <=). Логические операторы (and, or, not). Конструкция if-elif-else: синтаксис, отступы.

Практика. Решение задач: определение чётности числа, сравнение трёх чисел, игра «Угадай число» (одна попытка).

1.4. Вложенные условия. Тернарный оператор

Теория. Вложенные условные конструкции: когда они нужны и как их избежать. Конструкция match-case (Python 3.10+). Тернарный оператор: синтаксис и примеры использования.

Практика. Решение задач: определение диапазона чисел, классификация оценок по пятибалльной шкале, определение времени года по номеру месяца.

1.5. Цикл for. Функция range()

Теория. Понятие цикла. Цикл for: синтаксис, итерация по последовательностям. Функция range(). Break и continue.

Практика. Решение задач: вычисление суммы чисел от 1 до n, факториал числа, вывод таблицы умножения для одного числа.

1.6. Цикл while. Вложенные циклы

Теория. Цикл while: синтаксис, условие продолжения, опасность бесконечного цикла. Сравнение for и while. Вложенные циклы: принцип работы, сложность алгоритмов.

Практика. Решение задач: числа Фибоначчи, поиск простых чисел в диапазоне, полная таблица умножения (вложенный цикл).

1.7. Списки: создание, индексация, срезы

Теория. Списки как упорядоченные изменяемые коллекции. Создание списка (квадратные скобки, list()). Индексация (положительные и отрицательные индексы). Срезы (slice): синтаксис [start:stop:step], копирование списка через срез.

Практика. Решение задач: обработка списка оценок (среднее, максимум, минимум), реверс списка (без reverse), замена элементов по условию.

1.8. Методы списков: sort, reverse. List comprehensions

Теория. Методы списков: append, extend, insert, remove, pop, index, count, sort, reverse. Отличие sort() от sorted. List comprehensions (генераторы списков): синтаксис, условия в генераторах, вложенные генераторы.

Практика. Решение задач: фильтрация чётных чисел, генерация списка квадратов, сортировка строк по длине.

1.9. Словари и кортежи

Теория. Кортежи: неизменяемые последовательности, создание, отличия от списков. Словари (dict): структура «ключ–значение», создание, доступ по ключу, методы keys(), values(), items(). Итерация по словарю.

Практика. Решение задач: телефонная книга (добавление, поиск, удаление контактов), подсчёт частоты слов в тексте, простой переводчик.

1.10. Вложенные словари. Dict comprehensions

Теория. Вложенные словари: структура, доступ к элементам, итерация. Dict comprehensions (генераторы словарей): синтаксис, примеры. Метод get() для безопасного доступа.

Практика. Решение задач: база данных учеников (имя → {возраст, оценки}), группировка по возрасту, агрегация оценок (средний балл по каждому ученику).

1.11. Функции: def, return, параметры

Теория. Определение функции (def). Параметры и аргументы. Return — возврат значения. Функции без return (None). Область видимости переменных (локальные vs глобальные). Global.

Практика. Написание функций: среднее арифметическое списка, максимум из двух чисел, проверка числа на простоту.

1.12. Параметры по умолчанию. *args, **kwargs

Теория. Параметры по умолчанию (default arguments). Именованные аргументы. Переменное количество аргументов: *args (кортеж) и **kwargs (словарь).

Практика. Написание универсальной функции суммирования (любого количества чисел). Функция логирования с параметрами по умолчанию.

1.13. Lambda-функции. map, filter

Теория. Анонимные функции (lambda): синтаксис, ограничения. Функции высшего порядка: map (применение функции к каждому элементу), filter (отбор элементов по условию).

Практика. Применение lambda для сортировки списка словарей по ключу. Фильтрация и преобразование данных с помощью map и filter.

1.14. Введение в Git.

Теория. Что такое Git, зачем нужен контроль версий. Установка Git, базовая настройка (user.name, user.email).

Практика. Установка, настройка, проверка работы.

1.15. Базовые команды

Теория. Инициализация репозитория, отслеживание файлов, создание коммитов. Базовые команды: init, add, commit, status, log

Практика. Создание локального репозитория, добавление файлов, коммиты, просмотр истории.

1.16. Ветвление: branch, checkout, merge

Теория. Понятие веток. Создание, переключение, слияние веток.

Практика. Создание новой ветки, работа в ней, слияние с основной.

1.17. Работа с удалённым репозиторием

Теория. Регистрация на удаленном репозитории. remote, push, pull, clone.

Практика. Связывание локального репозитория с удаленным репозиторием, push/pull.

1.18. .gitignore, разрешение конфликтов

Теория. Файл .gitignore. Конфликты при слиянии и их разрешение.

Практика. Создание .gitignore, искусственный конфликт и его разрешение.

1.19. Отладка и обработка ошибок в Python.

Теория. Типы ошибок в Python: SyntaxError, NameError, TypeError, ValueError, IndexError, KeyError, AttributeError. Чтение traceback (стек вызовов). Простая отладка с помощью print(). Конструкция try-except для перехвата исключений. Блоки else и finally.

Практика. Разбор и исправление кода с разными типами ошибок. Написание небольших программ с try-except (деление на ноль, преобразование типов, работа со словарём).

2. NumPy, Pandas, визуализация, EDA

2.1. NumPy: массивы, shape, reshape

Теория. Библиотека NumPy. Массивы (ndarray), атрибуты shape, ndim, dtype, size.

Практика. Создание одномерных и двумерных массивов. Изменение формы reshape. Базовые операции (сложение, умножение на число).

2.2. NumPy: срезы, арифметика, broadcasting

Теория. Индексация и срезы в многомерных массивах. Векторные операции (поэлементные). Broadcasting: правило распространения операций на массивы разной формы.

Практика. Векторные вычисления (скалярное произведение). Работа со срезами. Решение простых линейных уравнений (методом подбора).

2.3. Pandas: Series и DataFrame. Чтение CSV

Теория. Библиотека Pandas: назначение. Series (одномерный массив с индексом). DataFrame (таблица). Чтение данных из CSV: pd.read_csv(), параметры sep, encoding.

Практика. Загрузка реального датасета (погода / цены на жилье или подобное). Просмотр первых/последних строк (head, tail). Информация о датасете (info).

2.4. Pandas: head, info, describe, фильтрация

Теория. Базовые методы DataFrame: head(), tail(), info(), describe(). Основы фильтрации: булевы маски, операторы & и |, метод query().

Практика. Отбор строк по условию (например, цены > 1000). Удаление ненужных колонок (drop). Анализ статистик (describe).

2.5. Pandas: groupby, agg

Теория. Группировка данных (groupby). Агрегация: agg(), apply(). Встроенные агрегатные функции (sum, mean, count, min, max).

Практика. Группировка по категориям: средняя цена по категориям товаров. Подсчёт количества элементов в каждой группе.

2.6. Pandas: merge, concat, pivot_table

Теория. Объединение таблиц: merge (по ключу, типы join: inner, left, right, outer), concat (вертикальное/горизонтальное). Сводные таблицы: pivot_table.

Практика. Объединение двух таблиц по общему столбцу. Создание сводной таблицы (средняя цена по категориям и по годам).

2.7. Очистка данных: пропуски (dropna, fillna)

Теория. Типы пропущенных значений (NaN, None, пустая строка). Методы работы с пропусками: dropna (удаление), fillna (заполнение). Выбор стратегии заполнения (среднее, медиана, мода, константа).

Практика. Обработка «грязного» датасета: заполнение пропусков средним/медианой. Удаление строк с пропусками в ключевых колонках.

2.8. Очистка данных: дубликаты, выбросы (IQR)

Теория. Дубликаты: duplicated(), drop_duplicates(). Выбросы. Межквартильный размах (IQR): Q1, Q3, формула границ (Q1–1,5*IQR, Q3+1,5*IQR). Z-оценка (для сильных).

Практика. Удаление дубликатов. Фильтрация выбросов по IQR.

2.9. Визуализация: Matplotlib

Теория. Библиотека matplotlib: pyplot. Типы графиков: plot, scatter, bar.

Практика. Построение линейного графика, диаграммы рассеяния и столбчатой диаграммы.

2.10. Визуализация: Subplots, настройка графиков

Теория. **Subplots.** Настройка: заголовки, подписи осей, легенда, сетка, стили.

Практика. Создание составного графика (2×2). Настройка внешнего вида.

2.11. Визуализация: Seaborn

Теория. Библиотека seaborn: статистическая визуализация. Основные функции: heatmap (тепловая карта корреляций), pairplot (матрица парных графиков), boxplot (коробчатая диаграмма)..

Практика. Матрица корреляций и её визуализация. Построение парных графиков.

2.12. Визуализация: Seaborn (распределения, сравнение выборок)

Теория. Violinplot (скрипичная диаграмма), kdeplot (оценка плотности распределения). Анализ распределений.

Практика. Анализ распределений, сравнение выборок, выявление выбросов.

2.13. Проект EDA - разведочный анализ данных.

Практика. Выбор датасета, загрузка, первичный осмотр, постановка вопросов. Очистка данных, построение 4+ графиков, формулировка выводов. Презентация результатов.

2.14. Самостоятельный разведочный анализ нового датасета.

Практика. Самостоятельный EDA нового датасета

3. Введение в ML

3.1. Что такое ML. Типы задач

Теория. Определение машинного обучения. Отличие ML от традиционного программирования. Типы задач: обучение с учителем (регрессия, классификация), обучение без учителя (кластеризация, ассоциативные правила), обучение с подкреплением. Примеры задач из реальной жизни.

Практика. Разбор кейсов: прогнозирование цены (регрессия), спам-фильтр (классификация), группировка покупателей (кластеризация).

3.2. Линейная регрессия (1 признак). MSE

Теория. Линейная регрессия: модель y = wx + b. Функция потерь — среднеквадратичная ошибка (MSE). Визуализация линии регрессии.

Практика. Расчёт MSE вручную. Построение линии регрессии по точкам.

3.3. sklearn: LinearRegression

Теория. Библиотека scikit-learn (sklearn). Класс LinearRegression: методы fit (обучение), predict (предсказание), coef_, intercept_.

Практика. Обучение модели на одном признаке. Предсказание для новых точек. Оценка качества на тестовых данных.

3.4. Множественная регрессия. R^2

Теория. Множественная линейная регрессия (несколько признаков). Коэффициент детерминации R^2: смысл, интерпретация, формула (1 - остаточная_дисперсия / общая_дисперсия). Сравнение моделей по R^2.

Практика. Обучение на нескольких признаках. Вычисление R^2. Сравнение разных моделей с разным набором признаков.

3.5. Регуляризация Ridge и Lasso (обзор)

Теория. Проблема переобучения при большом количестве признаков. Идея регуляризации: штраф за большие коэффициенты.

Практика. Сравнение обычной, Ridge и Lasso регрессии на одном датасете с большим количеством признаков.

3.6. Градиентный спуск (интуиция)

Теория. Идея градиентного спуска: поиск минимума функции потерь. Понятие градиента (направление наискорейшего возрастания). Learning rate (скорость обучения) — как его выбирать. Отличие batch, mini-batch и stochastic GD.

Практика. Визуализация сходимости градиентного спуска для разных learning rate (слишком маленький/большой/оптимальный).

3.7. Реализация стохастического градиентного спуска на numpy

Практика. Написание своего градиентного спуска для 1 признака (без sklearn). Сравнение коэффициентов w и b с sklearn. Визуализация уменьшения loss на каждой итерации.

3.8. Логистическая регрессия: сигмоида

Теория. Логистическая регрессия — модель для бинарной классификации. Сигмоидная функция σ(z) = 1/(1+e^{-z}): свойства, перевод линейной комбинации в вероятность. Порог принятия решения.

Практика. Визуализация сигмоиды. Классификация простого датасета (2 признака — рост и вес) на пол (мужской/женский).

3.9. sklearn: LogisticRegression. Log loss

Теория. Функция потерь для логистической регрессии: логистическая потеря (log loss / binary cross-entropy). Интерпретация: чем увереннее модель, тем ниже loss.

Практика. Обучение классификатора на датасете. Предсказание вероятностей (predict_proba). Визуализация границы принятия решения.

3.10. Метрики: confusion matrix, accuracy

Теория. Матрица ошибок (confusion matrix): TP, TN, FP, FN. Accuracy (доля правильных ответов). Когда accuracy неинформативна (несбалансированные классы).

Практика. Расчёт accuracy. Построение матрицы ошибок с помощью seaborn (тепловая карта).

3.11. Метрики: precision, recall, F1

Теория. Precision (точность) — доля релевантных среди предсказанных. Recall (полнота) — доля найденных релевантных. F1-мера — гармоническое среднее. Когда какая метрика важна.

Практика. Расчёт на несбалансированном датасете (например, редкое заболевание). Анализ ошибок модели.

3.12. Метрики: ROC-кривая и AUC

Теория. ROC-кривая (Receiver Operating Characteristic): зависимость True Positive Rate от False Positive Rate. AUC (Area Under Curve) — площадь под ROC-кривой. Интерпретация AUC: 0,5 — случайное угадывание, 1,0 — идеальная модель.

Практика. Построение ROC-кривой (sklearn.metrics.roc_curve). Расчёт AUC. Сравнение двух моделей.

3.13. Train-test split

Теория. Зачем делить данные на обучающую и тестовую выборки. Проблема утечки данных (data leakage). Стратифицированное разделение (stratify) для несбалансированных классов.

Практика. Разделение датасета (train_test_split). Обучение на train, проверка на test. Сравнение качества на train и test.

3.14. Переобучение и недообучение

Теория. Недообучение (underfitting). Переобучение (overfitting). Методы борьбы: регуляризация, ранняя остановка, увеличение данных.

Практика. Разделение датасета (train_test_split). Обучение на train, проверка на test. Сравнение качества на train и test. Эксперименты со сложностью модели (полиномиальная регрессия). Построение learning curves.

4. Деревья решений, случайный лес, кластеризация.

4.1. Деревья решений: как работает эта модель

Теория. Дерево решений — модель для классификации и регрессии. Структура: корень, узлы-условия, листья-ответы. Разбиение пространства признаков на прямоугольные области. Критерии выбора атрибута: Джини (Gini impurity), энтропия (Information gain).

Практика. Визуализация дерева на игрушечном датасете. Интерпретация правил (если признак <= X, то класс А, иначе Б).

4.2. Ограничение глубины, важность признаков

Теория. Проблема переобучения в деревьях. Ограничение глубины (max_depth). Минимальное количество образцов в листе (min_samples_leaf). Важность признаков (feature importance) — как её вычисляет дерево.

Практика. Эксперименты с глубиной дерева: сравнение точности на train и test. Определение важности признаков, визуализация.

4.3. Случайный лес: bagging, ансамбль

Теория. Ансамблевые методы: идея объединения нескольких моделей. Случайный лес (RandomForest) — ансамбль деревьев. Bagging (Bootstrap Aggregating): обучение каждого дерева на случайной подвыборке с возвращением. Случайный выбор признаков для каждого разбиения. OOB-ошибка (Out-Of-Bag) — оценка качества без тестовой выборки.

Практика. Сравнение одного дерева и случайного леса. Оценка OOB-ошибки.

4.4. Feature importance. Настройка леса

Теория. Важность признаков в случайном лесе (усреднение по деревьям). Гиперпараметры случайного леса: n_estimators (количество деревьев), max_depth, min_samples_split. Влияние на качество и время обучения.

Практика. Настройка параметров (n_estimators, max_depth). Визуализация важности признаков.

4.5. K-means: идея, алгоритм

Теория. Задача кластеризации: разделение данных на группы без меток. Алгоритм K-means: инициализация центроидов, назначение точек ближайшему центроиду, пересчёт центроидов. Критерий остановки.

Практика. Кластеризация игрушечных данных. Визуализация центроидов и областей.

4.6. Silhouette score. Выбор K

Теория. Выбор количества кластеров K. Метод локтя (elbow method). Silhouette score.

Практика. Выбор оптимального K через elbow метод и silhouette score. Визуализация кластеров (диаграмма рассеяния).

4.7. Иерархическая кластеризация

Теория. Иерархическая кластеризация: агломеративная (снизу вверх) и дивизивная (сверху вниз). Дендрограмма — визуализация иерархии. Выбор порога для отсечения.

Практика. Построение дендрограммы. Выбор порога, визуализация кластеров.

4.8. Масштабирование признаков

Теория. Зачем масштабировать признаки: влияние на алгоритмы, использующие расстояния (K-means, KNN, градиентный спуск). StandardScaler (Z-нормализация): (x — mean) / std. MinMaxScaler: (x — min) / (max — min).

Практика. Влияние масштабирования на K-means (до и после). Сравнение качества.

4.9. OneHotEncoding, LabelEncoding

Теория. Работа с категориальными признаками. LabelEncoding: преобразование категорий в числа (проблема — вносит порядок). OneHotEncoding: создание бинарных столбцов для каждой категории (избегает ложного порядка).

Практика. Преобразование категориальных признаков в числовые. Сравнение подхода Label и OneHot.

4.10. Pipeline: масштабирование → модель

Теория. Pipeline — объединение этапов предобработки и модели в один объект. Преимущества: удобство, предотвращение утечек данных. ColumnTransformer — применение разных трансформаций к разным колонкам.

Практика. Создание пайплайна для случайного леса (масштабирование → модель). Кросс-валидация пайплайна.

4.11. Введение в GridSearchCV.

Теория. Понятие гиперпараметров модели. Проблема ручного подбора. Библиотека sklearn.model_selection.GridSearchCV. Сетка параметров (param_grid). Кросс-валидация при подборе. Атрибуты best_params_, best_score_, best_estimator_.

Практика. Применение GridSearchCV для подбора гиперпараметров RandomForest (n_estimators, max_depth) на небольшом датасете. Сравнение качества до и после подбора

5. ФИНАЛЬНЫЙ ПРОЕКТ

5.1. Выбор темы финального проекта

Теория. Постановка задачи машинного обучения. Выбор целевой метрики. Критерии хорошего датасета. Типы задач: регрессия, классификация, кластеризация.

Практика. Поиск датасета (Kaggle, UCI Machine Learning Repository, OpenData). Формулировка задачи, выбор метрики качества. Создание репозитория на GitHub для проекта.

5.2. Проект: сбор и очистка данных

Практика. Загрузка данных из CSV/Excel/API. Первичный осмотр структуры данных. Обработка пропусков (dropna, fillna). Удаление дубликатов. Фильтрация выбросов (IQR или Z-оценка). Сохранение очищенного датасета. Коммит изменений в Git.

5.3. Проект: EDA для выбранного датасета

Практика. Построение графиков: гистограммы распределений, ящики с усами (boxplot), диаграммы рассеяния (scatter), матрица корреляций (heatmap). Анализ связей между признаками. Формулировка предварительных выводов. Коммит в Git.

5.4. Проект: обучение первой модели

Практика. Обучение LinearRegression (для регрессии) или LogisticRegression (для классификации). Оценка качества.

5.5. Проект: обучение второй модели (RandomForest)

Практика. Обучение RandomForest, настройка гиперпараметров (n_estimators, max_depth).

5.6. Проект: сравнение метрик, выбор лучшей модели

Практика. Сравнение моделей по метрикам. Визуализация результатов (сравнение предсказаний). Метрики для регрессии регрессии: MSE, R^2, MAE, для классификации: accuracy, precision, recall, F1, ROC-AUC. Сравнение результатов в таблице или на графике. Выбор лучшей модели по совокупности метрик. Обоснование выбора.

5.7. Проект: оформление ноутбука

Практика. Структурирование кода: разделение на ячейки с заголовками (Markdown). Добавление комментариев к сложным фрагментам. Формулировка выводов по каждому этапу (очистка, EDA, обучение, сравнение). Оформление финального отчёта в Jupyter Notebook. Коммит готового ноутбука в Git.

5.8. Подготовка презентации

Теория. Структура презентации проекта. Как отвечать на вопросы.

Практика. Создание слайдов, формулировка выступления.

5.9. **Защита проектов (групповая)**

Практика. Презентация проектов, ответы на вопросы.

5.10. **Защита проектов (индивидуальная)**

Практика. Индивидуальная защита проектов

5.11. Доработка проектов по замечаниям

Практика. Устранение ошибок, улучшение моделей, доработка презентаций.

6. КОНТРОЛЬНОЕ И ИТОГОВОЕ ЗАНЯТИЯ

6.1. Контрольное занятие

Практика. Промежуточный контроль: Выполнение практической работы.

6.2. Итоговое занятие

Практика Рефлексия. Итоговый контроль: защита проекта

МЕТОДИЧЕСКИЕ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Учебно-методический комплекс образовательной программы «Машинное обучение и анализ больших данных» состоит из следующих компонентов:

1. Описания применяемых педагогических методик и технологий.
2. Системы средств обучения (методико-прикладные средства: учебно-методические материалы, дидактические средства и т.п.).
3. Информационных источников, используемых при реализации программы.
4. Системы средств контроля результативности обучения (описание форм и средств выявления, фиксации и предъявления результатов обучения, а также их периодичности).

1. ПРИМЕНЯЕМЫЕ ПЕДАГОГИЧЕСКИЕ МЕТОДИКИ И ТЕХНОЛОГИИ

В образовательной программе «Машинное обучение и анализ больших данных» используются как традиционные педагогические технологии, так и современные: технология индивидуализации, групповые технологии, технология исследовательского (проблемного) обучения, игровые технологии, информационные и мультимедийные технологии, технология проектного обучения, здоровьесберегающие технологии.

Технология индивидуализации обучения (адаптивная), в основе которой лежит индивидуальный подход и индивидуальная форма обучения. Для развития навыков творческой работы учащихся на различных этапах обучения применяется дифференцированный подход к учащимся, что позволяет педагогу полнее учитывать индивидуальные возможности и личностные особенности учащегося, достигать более высоких результатов в обучении и развитии творческих и аналитических способностей. В рамках данной программы это выражается в предоставлении учащимся заданий разного уровня сложности (базовый, продвинутый, «хард»).

Групповые технологии: одновременная работа со всей группой, работа в парах (парное программирование), групповая работа на принципах дифференциации при выполнении проектов EDA и финального проекта.

Технология исследовательского (проблемного) обучения. Педагог использует следующие педагогические приемы: подводит детей к противоречию и предлагает им найти способ его разрешения, излагает различные точки зрения на вопрос, предлагает рассмотреть явление с различных позиций, побуждает детей делать сравнения, обобщения, выводы, ставит проблемные вопросы, задачи, задает проблемные задания (например, «почему модель переобучилась?», «как выбрать оптимальное количество кластеров?»).

Игровые технологии. Педагог использует интеллектуальные, социальные, психологические виды деятельности, по характеру педагогического процесса (обучающие, тренировочные, познавательные, контролирующие, познавательные, развивающие, репродуктивные, творческие, коммуникативные и др.), по игровой методике (сюжетные, ролевые, деловые, имитационные и др.), по игровой среде (с предметом и без, настольные, комнатные, компьютерные и др.). В программе применяются внутригрупповые соревнования по качеству моделей и скорости решения задач.

Информационные и мультимедийные технологии. Педагог использует компьютер на занятиях как наглядное пособие (показ презентаций, фильмов, интерактивных блокнотов), тренажер (обучающие программы по Python), средство диагностики и контроля (автоматизированное тестирование), Интернет-технологии для поиска информации учащимися, компьютерные программы для моделирования, анализа данных и программирования (Jupyter Notebook, Google Colab, Yandex DataSphere, VS Code).

Применение мультимедийной технологии обусловлено содержанием программы, где результатом проекта является разработка собственного проекта по анализу данных и машинному обучению. Данная технология обогащает процесс обучения, позволяя сделать обучение более эффективным, вовлекая в процесс восприятия учебной информации большинство чувственных компонент учащегося. Данная технология объединяет в себе как традиционную, статическую, визуальную информацию (текст, графику), так и динамическую (видеофрагменты, анимацию процессов обучения моделей).

Технология проектного обучения. Используется технология защиты индивидуальных или групповых проектов. В ходе проектной деятельности учащиеся работают над созданием своих индивидуальных или групповых проектов (проект EDA, финальный проект по машинному обучению), которые представляют на контрольных занятиях и итоговой аттестации. На занятиях педагог превращается в куратора или консультанта: помогает обучаемым в поиске источников и датасетов, сам является источником информации, поддерживает и поощряет учащихся, координирует и корректирует весь процесс, поддерживает непрерывную обратную связь.

Здоровьесберегающие технологии применяются педагогом с целью формирования у детей необходимых знаний и навыков по здоровому образу жизни, использование учащимися полученных знаний в повседневной жизни. На занятиях проводятся физкультминутки, динамические паузы и минутки релаксации, гимнастика для глаз с целью профилактики сколиоза, укрепления мышечного корсета, коррекции недостатков осанки и снижения зрительного утомления при длительной работе за компьютером.

.

2. СИСТЕМА СРЕДСТВ ОБУЧЕНИЯ

Блок методико-прикладных средств

1. Учебно-методические материалы:

- Справочная литература для педагога и учащихся по Python, Pandas, Scikit-learn;
- Планы конспектов занятий к разделам образовательной программы;
- Методические рекомендации по выполнению проектов EDA и финального проекта

2. Дидактические средства:

2.1. Дидактические пособия (раздаточный материал, вопросы и задания для устного или письменного опроса, тесты, практические задания, упражнения к различным разделам программы, карточки с задачами разного уровня сложности).

2.2. Авторские электронные образовательные ресурсы:

– авторские компьютерные презентации к различным разделам программы (введение в ML, линейная регрессия, деревья решений, случайный лес, кластеризация);

– учебные видеоуроки и/или интерактивные Jupyter Notebook, созданные педагогом, по следующим темам: «Введение в Python», «Условные операторы и циклы», «Списки и словари», «Функции и lambda-выражения», «NumPy и векторные вычисления», «Pandas: работа с данными», «Визуализация с matplotlib и seaborn», «Разведочный анализ данных», «Линейная регрессия», «Логистическая регрессия», «Метрики качества моделей», «Деревья решений и случайный лес», «Кластеризация методом K-means», «Инженерия признаков и пайплайны»;

- контрольные тесты по ключевым темам программы;

2.3. Диагностические и контрольные материалы:

- входное тестирование (письменная часть + устная беседа);
- практическая работа по разделу 1;
- практическая работа по разделу;
- практическая работа по разделу 3;
- Практическая работа по разделу 4;
- критерии оценки финального проекта (ноутбук + код + презентация + защита);
- карта результативности освоения образовательной программы;
- карта наблюдения «Метапредметные и личностные результаты»;
- информационная карта участия и результативности учащихся в конкурсных мероприятиях.

3. ИНФОРМАЦИОННЫЕ ИСТОЧНИКИ

Список литературы для педагога:

1. Бхаргава, А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих / А. Бхаргава. — Санкт-Петербург : Питер, 2018. — 288 с. — ISBN 978-5-496-02536-0. — Текст: непосредственный.

2. Вандер Плас, Дж. Python для сложных задач: наука о данных и машинное обучение / Дж. Вандер Плас. — Санкт-Петербург : Питер, 2018. — 576 с. — ISBN 978-5-496-03068-5. — Текст: непосредственный.

3. Гéрон, О. Прикладное машинное обучение с помощью Scikit-Learn, Keras и TensorFlow / О. Гéрон. — 3-е изд. — Санкт-Петербург : Питер, 2023. — 896 с. — ISBN 978-5-4461-2051-2. — Текст: непосредственный.

4. Лутц, М. Изучаем Python / М. Лутц. — 5-е изд. — Санкт-Петербург : Питер, 2019. — 1328 с. — ISBN 978-5-496-01548-4. — Текст: непосредственный.

5. Маккини, У. Python и анализ данных / У. Маккини. — Москва : ДМК Пресс, 2020. — 664 с. — ISBN 978-5-97060-374-1. — Текст: непосредственный.

6. Мюллер, А. Введение в машинное обучение с помощью Python / А. Мюллер, С. Гвидо. — Москва : Диалектика, 2017. — 480 с. — ISBN 978-5-9909446-9-7. — Текст: непосредственный.

7. Рашка, С. Python и машинное обучение / С. Рашка, В. Мирджалили. — 3-е изд. — Москва : ДМК Пресс, 2020. — 698 с. — ISBN 978-5-97060-822-7. — Текст: непосредственный.

8. Хэдли, У. Основы Data Science и Big Data. Python и наука о данных / У. Хэдли. — Москва : Вильямс, 2020. — 336 с. — ISBN 978-5-8459-1812-5. — Текст: непосредственный.

Список литературы для учащихся и их родителей:

1. Бэрри, П. Изучаем программирование на Python / П. Бэрри. — Москва : Эксмо, 2019. — 384 с. — ISBN 978-5-04-098464-4. — Текст: непосредственный.

2. Свейгарт, Э. Автоматизация рутинных задач с помощью Python / Э. Свейгарт. — Санкт-Петербург : Питер, 2019. — 576 с. — ISBN 978-5-4461-0869-5. — Текст: непосредственный.

3. Трофимов, П. А. Python для детей. Самоучитель по программированию / П. А. Трофимов. — Москва : ДМК-Пресс, 2020. — 264 с. — ISBN 978-5-97060-812-8. — Текст: непосредственный.

Интернет-источники

4. Kaggle: платформа для соревнований по Data Science. — URL: https://www.kaggle.com/ — Режим доступа: свободный. — Текст: электронный.

5. Stepik: образовательная платформа. Курс «Поколение Python». — URL: https://stepik.org/course/58852 — Режим доступа: свободный. — Текст: электронный.

6. Yandex DataSphere: облачная среда для разработки в области Data Science. — URL: https://yandex.cloud/ru/datasphere — Режим доступа: свободный. — Текст: электронный.

7. UCI Machine Learning Repository: архив датасетов для машинного обучения. — URL: https://archive.ics.uci.edu/ — Режим доступа: свободный. — Текст: электронный.

4. СИСТЕМЫ СРЕДСТВ КОНТРОЛЯ РЕЗУЛЬТАТИВНОСТИ ОБУЧЕНИЯ

Для отслеживания результативности образовательной деятельности по программе проводятся: входной контроль, текущий контроль, промежуточная аттестация и итоговый контроль (итоговое оценивание).

Входной контроль проводится на начальном этапе формирования коллектива (сентябрь первого года обучения) с целью изучения отношения ребенка к выбранной деятельности и его способностей в этой области, личностных качеств ребенка в форме письменного тестирования и устной беседы. Для детей, которые принимаются в учебную группу на свободные места в течение учебного года, входной контроль также осуществляется в форме тестирования и собеседования.

Текущий контроль осуществляется на занятиях в течение всего учебного года для отслеживания уровня и качества освоения учебного материала программы отдельных тем или разделов программы в форме выполнения практической работы (написание кода, анализ данных) и/или собеседования.

Промежуточная аттестация в форме выполнения практической работы проводится в конце I полугодия (практическая работа №2) в целях изучения динамики освоения предметного содержания (Python, Pandas, EDA, визуализация), личностного развития, взаимоотношений в коллективе.

Итоговый контроль (итоговое оценивание) проводится по завершению всего периода обучения в конце 1 года для оценки уровня и качества освоения учащимися программы. Итоговый контроль проводится в форме защиты финального проекта (индивидуального или группового).

Возможные средства и формы фиксации результатов по программе: оценочные листы текущего, промежуточного и итогового контроля; ведомости учёта результатов контрольных работ; благодарности; видеозаписи защит проектов; грамоты, дипломы; отзывы (детей и родителей); протоколы конкурсов и/или соревнований; фотоотчеты; портфолио учащихся (ноутбуки с проектами) и т.п.

Возможные средства и формы предъявления результатов по программе: карта результативности освоения образовательной программы «Машинное обучение и анализ больших данных» (по модулям программы); карта наблюдения «Метапредметные и личностные результаты», информационная карта участия и результативности учащихся образовательной программы «Машинное обучение и анализ больших данных» в конкурсных мероприятиях; видеозаписи защит проектов; фотоотчеты; публикации в социальных сетях и на сайте учреждения; участие в научно-практических конференциях и хакатонах и т.п.

ФОРМЫ ФИКСАЦИИ РЕЗУЛЬТАТОВ ПО ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЕ «МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ БОЛЬШИХ ДАННЫХ»

Оценочный лист промежуточной аттестации за 1 полугодие (декабрь)

По образовательной программе: **«МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ БОЛЬШИХ ДАННЫХ»**

Задание: Выполнение практической работы: анализ нового датасета (EDA) с использованием Python, Pandas, визуализации..

№ п/п	Фамилия, имя учащегося	Критерии оценки			**Средний балл**
		Владение Python и Git	**Владение Pandas и визуализацией**	**Умение проводить EDA и делать выводы**	
		от 1 до 3 баллов	от 1 до 3 баллов	от 1 до 3 баллов	от 1 до 3 баллов
1.					
2.					
...					
Средний балл:					

Оценка каждого критерия:

Баллы	Уровень	Описание
3	Высокий (B)	Самостоятельно выполняет комплексную работу, код чистый, выводы обоснованы
2	Средний (C)	Учащийся выполняет задания с незначительными ошибками или помощью педагога
1	Минимальный (M)	Учащийся испытывает серьёзные затруднения, требуется постоянная помощь

Оценочный лист итогового контроля (май)

По образовательной программе: **«МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ БОЛЬШИХ ДАННЫХ»**

Задание: Защита финального проекта (индивидуального или группового)

№ п/п	Фамилия, имя учащегося	Критерии оценки			**Средний балл**
		Качество EDA и очистки данных	**Работоспособность и сравнение моделей**	**Оформление ноутбука и презентации, защита**	
		от 1 до 3 баллов	от 1 до 3 баллов	от 1 до 3 баллов	от 1 до 3 баллов
1.					
2.					
...					
Средний балл:					

Оценка каждого критерия:

Баллы	Уровень	Описание
3	Высокий (В)	Проект полностью самостоятельно, код чистый, модели работают, защита уверенная
2	Средний (С)	Проект выполнен с незначительными ошибками или помощью педагога, защита удовлетворительная
1	Минимальный (М)	Проект не завершён или выполнен с грубыми ошибками, защита слабая

Оценочный лист текущего контроля
по образовательной программе «МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ БОЛЬШИХ ДАННЫХ»

Раздел образовательной программы: **1. Основы Python и Git**

Задание: Практическая работа — решение задач на пройденный материал

№ п/п	Фамилия, имя учащегося	Критерии оценки			**Средний балл**
		Уровень освоения синтаксиса и базовых конструкций Python	**Уровень владения Git**	**Умение отлаживать и обрабатывать ошибки**	
		от 1 до 3 баллов	от 1 до 3 баллов	от 1 до 3 баллов	от 1 до 3 баллов
1.					
2.					
...					
Средний балл:					

Оценка каждого критерия:

Баллы	Уровень	Описание
3	Высокий (В)	Учащийся самостоятельно и без ошибок пишет программы, использует Git, отлаживает код
2	Средний (С)	Учащийся выполняет задания с незначительными ошибками или помощью педагога
1	Минимальный (М)	Учащийся испытывает серьёзные затруднения, требуется постоянная помощь

Оценочный лист текущего контроля
по образовательной программе «МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ БОЛЬШИХ ДАННЫХ»

Раздел образовательной программы: **2. NumPy, Pandas, визуализация, EDA**

Задание: Практическая работа. Проект EDA — анализ выбранного датасета.

№ п/п	Фамилия, имя учащегося	Критерии оценки			**Средний балл**
		Умение загружать и очищать данные (Pandas)	**Умение визуализировать данные (matplotlib/seaborn)**	**Умение формулировать выводы по EDA**	
		от 1 до 3 баллов	от 1 до 3 баллов	от 1 до 3 баллов	от 1 до 3 баллов
1.					
2.					
...					
Средний балл:					

Оценка каждого критерия:

Баллы	Уровень	Описание
3	Высокий (В)	Самостоятельно загружает, очищает, визуализирует данные и делает глубокие выводы
2	Средний (С)	Учащийся выполняет задания с незначительными ошибками или помощью педагога
1	Минимальный (М)	Учащийся испытывает серьёзные затруднения, требуется постоянная помощь

Оценочный лист текущего контроля
по образовательной программе «МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ БОЛЬШИХ ДАННЫХ»

Раздел образовательной программы: **3. Введение в ML. Регрессия и классификация**

Задание: Практическая работа— сравнение LinearRegression и LogisticRegression на одном датасете

№ п/п	Фамилия, имя учащегося	Критерии оценки			**Средний балл**
		Понимание теории ML (типы задач, регрессия, классификация)	**Умение обучать и оценивать модели**	**Умение интерпретировать метрики**	
		от 1 до 3 баллов	от 1 до 3 баллов	от 1 до 3 баллов	от 1 до 3 баллов
1.					
2.					
...					
Средний балл:					

Оценка каждого критерия:

Баллы	Уровень	Описание
3	Высокий (В)	Самостоятельно обучает модели, выбирает метрики, интерпретирует результаты
2	Средний (С)	Учащийся выполняет задания с незначительными ошибками или помощью педагога
1	Минимальный (М)	Учащийся испытывает серьёзные затруднения, требуется постоянная помощь

Оценочный лист текущего контроля
по образовательной программе «МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ БОЛЬШИХ ДАННЫХ»

Раздел образовательной программы: **4. Деревья, лес, кластеризация**

Задание: Практическая работа— обучить RandomForest и K-means на одном датасете

№ п/п	Фамилия, имя учащегося	Критерии оценки			**Средний балл**
		Умение настраивать RandomForest и K-means	**Умение выбирать количество кластеров**	**Умение применять пайплайны и инженерию признаков**	
		от 1 до 3 баллов	от 1 до 3 баллов	от 1 до 3 баллов	от 1 до 3 баллов
1.					
2.					
...					
Средний балл:					

Оценка каждого критерия:

Баллы	Уровень	Описание
3	Высокий (В)	Самостоятельно обучает и настраивает модели, интерпретирует результаты
2	Средний (С)	Учащийся выполняет задания с незначительными ошибками или помощью педагога
1	Минимальный (М)	Учащийся испытывает серьёзные затруднения, требуется постоянная помощь

ФОРМЫ ПРЕДЪЯВЛЕНИЯ РЕЗУЛЬТАТОВ ПО ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЕ
«МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ БОЛЬШИХ ДАННЫХ»

Карта результативности освоения образовательной программы «МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ БОЛЬШИХ ДАННЫХ»
за 2026/2027 учебный год

№ п/п	Фамилия, имя учащегося	Раздел программы						Личный рейтинг учащегося за (средний балл)	Уровень освоения образова-тельной программы (М/С/В)
		1. Основы Python и Git	2. Pandas, EDA, визуализация	3. Регрессия и классификация	4. Деревья, лес, кластеризация	6. Контрольное и итоговое занятия			
						Промежуточная аттестация (I полугодие) —	Итоговый контроль (финальный проект)		
1.									
2.									
…									
Средний балл по разделу:									

Информационная карта участия и результативности учащихся
образовательной программы «МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ БОЛЬШИХ ДАННЫХ» в конкурсных мероприятиях
за _ полугодие 20__/20__ учебного года

№ п/п	Фамилия, имя учащегося	На уровне учреждения			На уровне района			На уровне города			На международном и российском уровне			Итого баллов
		Участие (выступление)	Призер, дипломант	Победитель	Участие (выступление)	Призер, дипломант	Победитель	Участие (выступление)	Призер, дипломант	Победитель	Участие (выступление)	Призер, дипломант	Победитель	
		1 б	**2 б**	**3 б**	**2 б**	**3 б**	**4 б**	**3 б**	**4 б**	**5 б**	**4 б**	**5 б**	**6 б**	
1.														
…														
Общая сумма баллов по группе: ___														

Карта наблюдения «Метапредметные и личностные результаты»
по образовательной программе «МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ БОЛЬШИХ ДАННЫХ»
за _ полугодие 20__/20__ учебного года

№ п/п	Фамилия, имя учащегося	Метапредметные результаты								Личностные результаты				Итого (средний показатель)	
		Коммуникативная компетентность в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности		Внимание, память, наблюдательность (согласно возрасту)		Воображение, творческие способности, стремление к достижению поставленной цели (согласно возрасту)		Логическое мышление и аналитические способности (согласно возрасту)		Ответственность за свою работу		Мотивация к занятиям программированием			
		Полугодие													
		1-е	*2-е*	*1-е*	*2-е*	*1-е*	*2-е*	*1-е*	*2-е*	*1-е*	*2-е*	*1-е*	*2-е*	*1-е*	*2-е*
1															
2															
…															
Итого среднее:															

Минимальный уровень – от 1 до 1,4 балла;
Средний уровень – от 1,5 до 2,4 баллов;
Высокий уровень – от 2,5 до 3 баллов.

Критерии определения степени выраженности оцениваемого качества

№ п/п	Показатели (оцениваемые параметры)	Критерии	Степень выраженности оцениваемого качества	Возможное количество баллов	Методы диагностики
Метапредметные результаты обучения по дополнительной программе					
1.	Коммуникативная компетентность в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности	Потребность в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности	Минимальный уровень (учащийся избегает сотрудничества со сверстниками, может провоцировать конфликты)	1 балл	Карта наблюдения
			Средний уровень (учащийся не всегда сотрудничает со сверстниками, иногда демонстрирует недоброжелательное отношение)	2 балла	
			Высокий уровень (учащийся инициативен в общих делах, доброжелательно относится к сверстникам)	3 балла	
2.	Внимание, память, наблюдательность (согласно возрасту)	Развитие внимания, памяти, наблюдательности (согласно возрасту)	Минимальный уровень (учащийся плохо запоминает пройденный материал, недостаточно внимателен и наблюдателен)	1 балл	Карта наблюдения
			Средний уровень (учащийся иногда не запоминает новые термины и последовательность действий, может быть недостаточно внимателен к происходящему и наблюдателен)	2 балла	
			Высокий уровень (учащийся обладает хорошей памятью и наблюдательностью, внимателен к происходящему)	3 балла	
3.	Воображение, творческие способности, стремление к достижению поставленной цели (согласно возрасту)	Развитие воображения, творческих способностей, стремления к достижению поставленной цели (согласно возрасту)	Минимальный уровень (учащийся практически не проявляет творческую инициативу и воображение, пассивно участвует в выполнении работы, требуется контроль педагога)	1 балл	Карта наблюдения
			Средний уровень (учащийся частично проявляет творческую инициативу и воображение (освоены элементы репродуктивной, имитационной деятельности), но не всегда самостоятелен в постановке задач)	2 балла	
			Высокий уровень (учащийся проявляет воображение, приобретен опыт самостоятельной творческой деятельности (оригинальность, индивидуальность, качественная завершенность результата), самостоятельно ставит перед собой задачи)	3 балла	
4.	Логическое мышление и аналитические способности (согласно возрасту)	Развитие способности логически мыслить и анализировать (согласно возрасту)	Минимальный уровень (учащийся испытывает серьезные затруднения при необходимости логического мышления, в процессе деятельности отсутствует анализ)	1 балл	Карта наблюдения
			Средний уровень (учащийся испытывает некоторые затруднения при необходимости логического мышления и/или анализе необходимых действий)	2 балла	

			Высокий уровень (учащийся занимается анализом проделанной работы, успешно выстраивает логические цепочки)	3 балла	
5.	Способность самостоятельно искать, анализировать, отбирать необходимую информацию	Самостоятельность в подборе и анализе информации	Минимальный уровень (учащийся испытывает серьезные затруднения при подборе и анализе информации, нуждается в постоянной помощи и контроле педагога)	1 балл	Анализ подбора информации
			Средний уровень (подбирает и анализирует информацию с самостоятельно и с помощью педагога или родителей)	2 балла	
			Высокий уровень (подбирает и анализирует информацию самостоятельно, не испытывает особых трудностей)	3 балла	
Личностные результаты обучения по дополнительной программе					
1.	Ответственность за свою работу	Способность ответственно относиться к учебе и своим поступкам	Минимальный уровень (учащийся не ощущает себя ответственным за обучение и свои поступки, не задумывается о последствиях, опаздывает на занятия)	1 балл	Карта наблюдения
			Средний уровень (ориентируется в правах и обязанностях учащихся, но не всегда выполняет возложенные обязанности, редко опаздывает на занятия)	2 балла	
			Высокий уровень (дисциплинирован, чувствует себя ответственным за порученное дело, иногда призывает окружающих к ответственности, выполняет порученное, без уважительной причины не задерживается)	3 балла	
2.	Мотивация к занятиям программированием	Устойчивость мотивации и интереса	Минимальный уровень (случайная, навязанная извне мотивация – родители, друзья. Неосознанный интерес, поддерживается педагогом)	1 балл	Карта наблюдения
			Средний уровень (неустойчивая мотивация, зависит от результативности занятий, интерес на уровне любознательности, иногда поддерживается самостоятельно)	2 балла	
			Высокий уровень (устойчивая мотивация, интерес на уровне увлечения, поддерживается самостоятельно)	3 балла	
3.	Положительное отношение к информатике и программированию	Устойчивость интереса к информатике и углубленному изучению программирования, машинного обучения, анализа данных	Минимальный уровень (учащийся не проявляет интереса к информатике и программированию)	1 балл	Опрос, анкетирование
			Средний уровень (учащийся не всегда проявляет положительное отношение и интерес к информатике и программированию)	2 балла	
			Высокий уровень (учащийся проявляет положительное отношение к информатике и интерес к углубленному изучению программирования и ML)	3 балла	