**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА   
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ**

**ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ – ФИЛИАЛ**

ОТЧЕТ

По направлению: Многомерный анализ данных

Выполнил студент группы Ик-731

Соколов Д.А

г. Нижний Новгород

2025

**Цель работы:** Познакомиться с понятием набора данных, их источниками и возможными представлениями (форматом).

**Задачи:**

1. Разобраться с содержанием понятия «набор данных» и целью их использования.
2. Найти в интернете источники данных, в том числе российские.
3. Скачать и проанализировать состав одного из найденных и заинтересовавших Вас набора данных.

Выполнение

Задание №1. Скачать и установить на свой девайс платформу Loginom

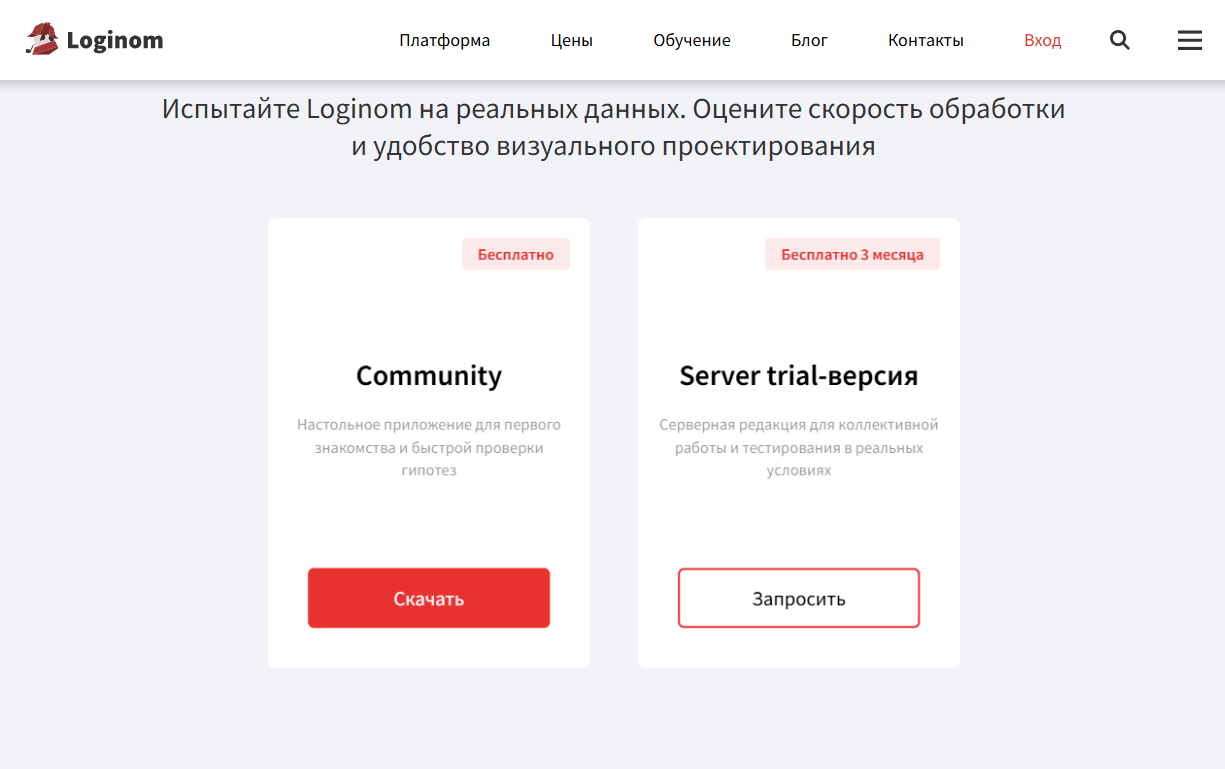


Рисунок 1 – Установка Loginom Community

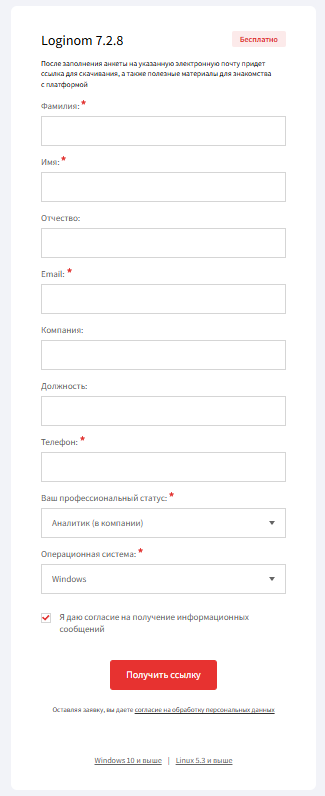


Рисунок 2 – Регистрация и получение установочного файла

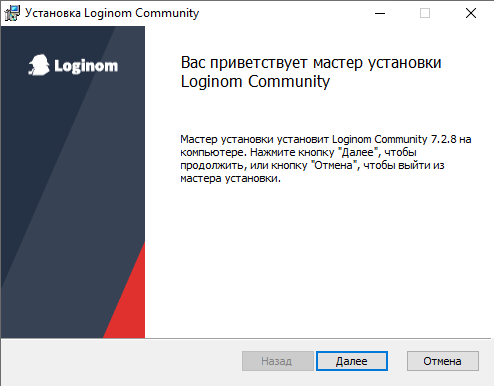


Рисунок 3 – Процесс установки

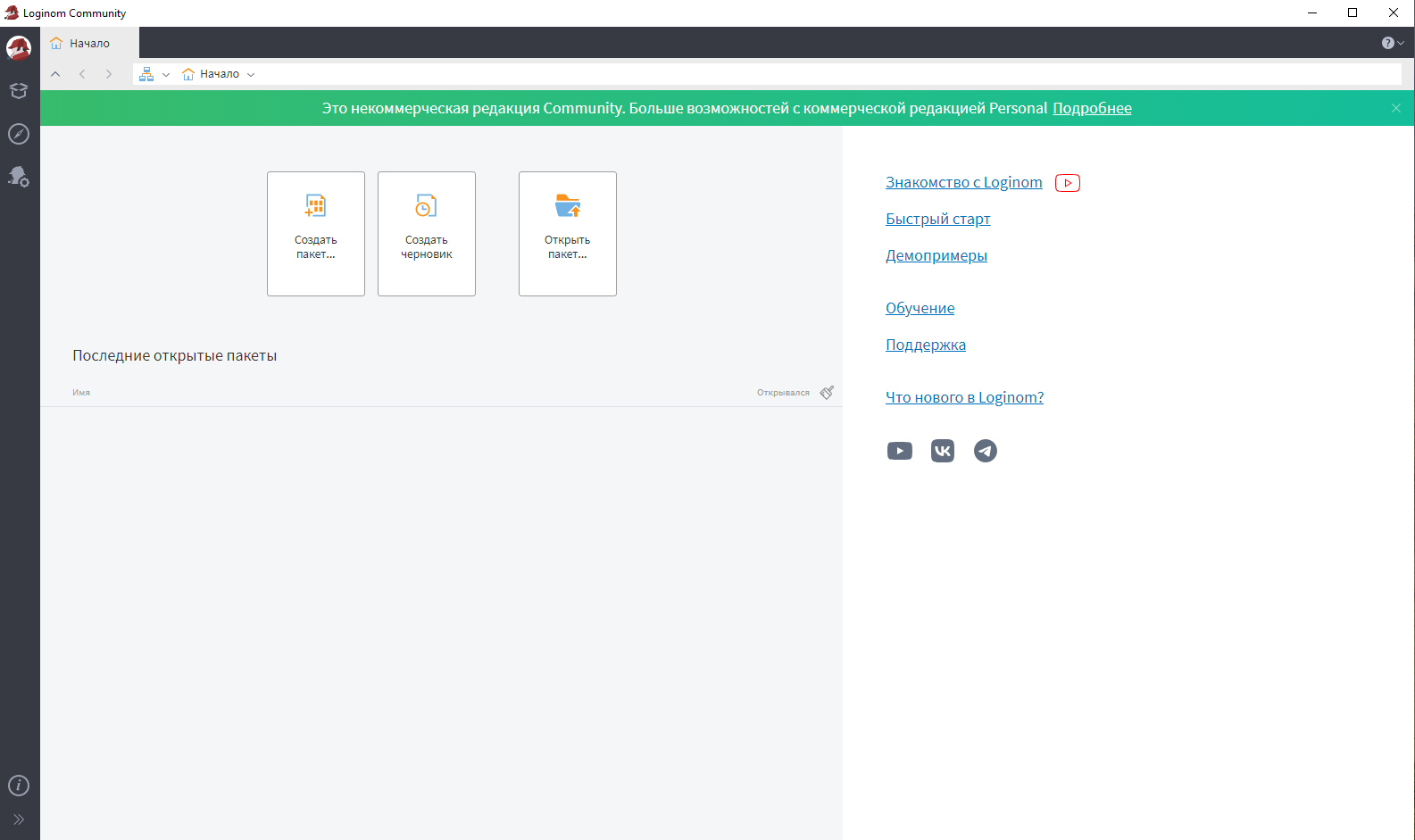


Рисунок 4 - Результат

Задание №2. Описать сферу деятельности по профилю вашей работы (вашего предприятия), в которой может потребоваться специалист по анализу данных.

Я работаю в организации АНОО Школа 800, на должности «Аналитик-программист».

Сфера моей деятельности - образование, а именно аналитическая поддержка работы образовательной организации на 3000 учеников. Здесь специалист по анализу данных необходим для решения следующих задач:

**Учебная аналитика**

1. анализ успеваемости и посещаемости учеников;
2. выявление закономерностей между нагрузкой, расписанием и результатами обучения;
3. прогнозирование вероятности академических задолженностей.

**Организационно-управленческая аналитика**

1. планирование нагрузки педагогов и распределение часов;
2. оценка эффективности учебных планов и расписаний;
3. анализ наполняемости классов и групп.

**Взаимодействие с родителями и администрацией**

1. построение дашбордов и отчетов для руководства;
2. подготовка аналитики по домашним заданиям, оценкам, вовлеченности учеников.

**Автоматизация и цифровизация процессов**

1. интеграция данных из разных источников (школьные API, базы данных, Excel-отчеты);
2. построение ETL-процессов и визуализаций в BI-системах (например, Metabase).

Таким образом, специалист по анализу данных помогает принимать управленческие решения на основе фактов, а не интуиции: от оптимизации расписания до стратегического планирования работы школы.

Задание №3. Ознакомиться со структурой предлагаемых наборов данных (по выбору).

Как пример данных я возьму измененную базу данных образовательно организации для аналитики данных.

**1. Обращения (support\_requests)**

id -уникальный идентификатор обращения (PK)

created\_at -дата и время создания

applicant\_id -ID заявителя (FK -users.id)

channel -канал обращения (телефон, email, чат, форма)

title -тема/заголовок

description -описание проблемы

category -категория (например: расписание, электронный дневник, доступ к системе)

priority -приоритет (низкий, средний, высокий)

status -статус (новая, в работе, ожидает ответа клиента, решена, закрыта)

closed\_at -дата и время закрытия

sla\_hors -время на решение

**2. Пользователи (users)**

id -уникальный ID пользователя (PK)

full\_name -ФИО

role -роль (ученик, родитель, учитель, администратор, сотрудник)

email -email

phone -телефон

created\_at -дата регистрации

**3. Ученики (students)**

id -уникальный ID ученика (PK)

user\_id -FK -users.id

class\_id -FK -classes.id

date\_of\_birth -дата рождения

enrolled\_at -дата поступления

**4. Классы (classes)**

id -уникальный ID класса (PK)

name -название (например, 10А, 11Б)

year -учебный год

class\_teacher\_id -FK -teachers.id (классный руководитель)

**5. Учителя (teachers)**

id -уникальный ID учителя (PK)

user\_id -FK -users.id

subject\_id -FK -subjects.id (основной предмет)

hire\_date -дата приема на работу

**6. Предметы (subjects)**

id -уникальный ID предмета (PK)

name -название предмета (Математика, История, Физика)

code -короткий код (MATH, HIST, PHYS)

**7. Расписание / Занятия (schedule)**

id -уникальный ID записи (PK)

class\_id -FK -classes.id

subject\_id -FK -subjects.id

teacher\_id -FK -teachers.id

date -дата занятия

lesson\_number -номер урока (1, 2, 3…)

room -аудитория

topic -тема урока

**Такая структура позволит:**

1. хранить обращения сотрудников/родителей и анализировать SLA;
2. связывать учеников, классы, учителей и предметы;
3. строить дашборды: нагрузка учителей, прогулы учеников, эффективность расписания;
4. делать прогнозы по успеваемости и загруженности.

Задание №4. Знать чёткое определения понятия «Набор данных» по ссылке

Я разобрал статью по ссылке и сделал выводы.

Dataset для машинного обучения – это обработанная и структурированная информация в табличном виде, другими словами это таблица с данными, которые уже упорядочены и подготовлены для анализа.

**Строки в этой таблице** – это объекты (например, конкретные люди, товары, транзакции).

Столбцы – это признаки (характеристики объектов).

**Признаки бывают двух типов:**

Независимые (предикторы) – данные, которые мы подаём на вход (например, возраст, доход, количество покупок).

Зависимые (целевые признаки) – то, что мы хотим предсказать (например, купит клиент товар или нет).

**В зависимости от задачи целевой признак может быть разным:**

0/1 (да/нет) – для бинарной классификации,

несколько меток – для мультиклассовой или мультимаркерной классификации,

числа – для регрессии (например, прогноз цены).

**Кроме того, у датасета есть разные подвыборки:**

1. обучающая – на ней учат модель,
2. тестовая – на ней проверяют качество,
3. валидационная – помогает выбрать наилучшую модель и её настройки.

Важно, чтобы эти выборки не пересекались, иначе модель "зазубрит" данные и будет плохо работать на новых примерах.

Итог: датасет – это структурированный набор данных (обычно в табличном виде), где каждая строка – объект, а каждый столбец – его характеристика. Он нужен, чтобы обучать и проверять модели машинного обучения, которые потом умеют прогнозировать или классифицировать новые объекты.

**Лабораторная работа №1**

4 Задание к практическим занятиям

4.1 Знакомство с системами Anaconda, Jupyter Notebook и Pandas.

Установить дистрибутив Anaconda.

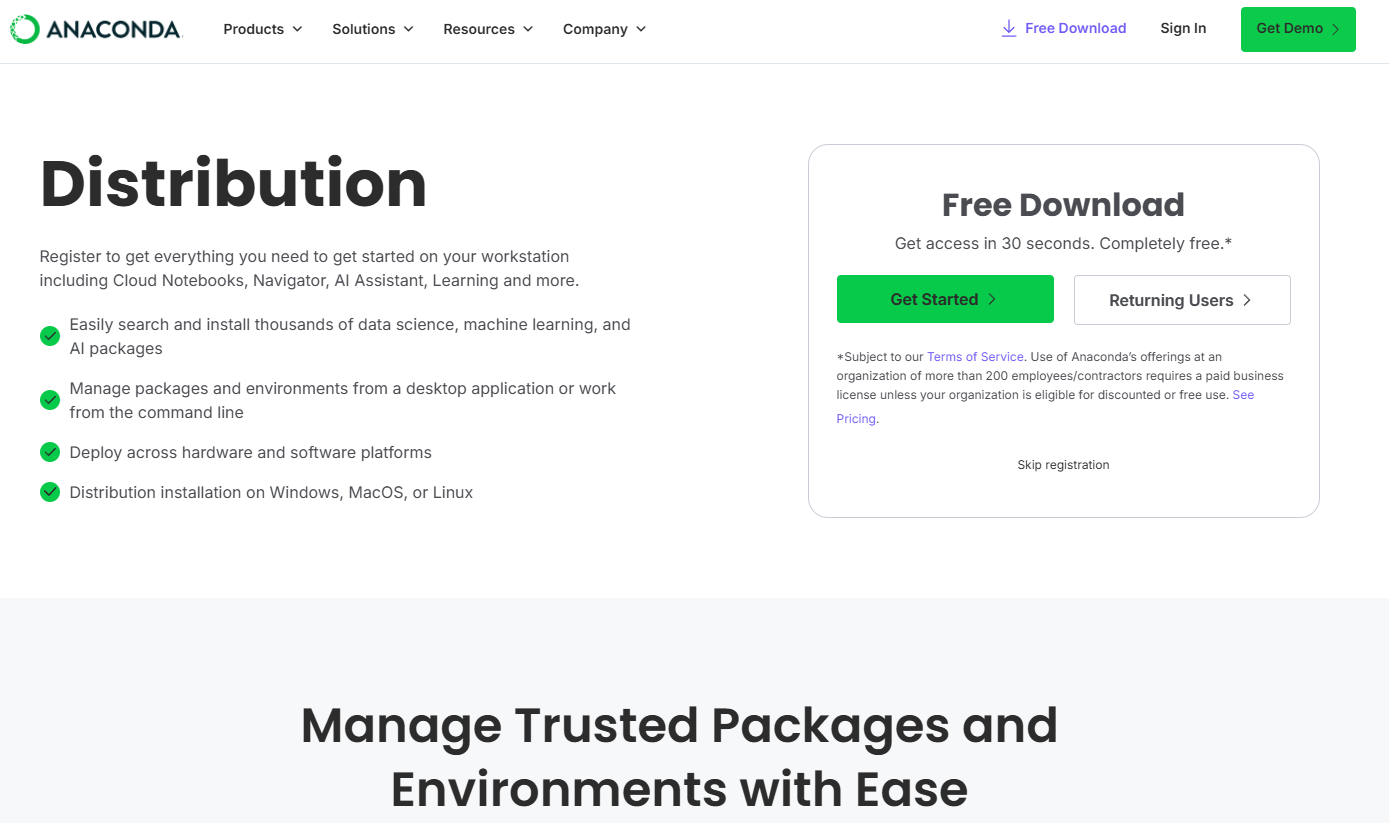


Рисунок 5 – Скачал с официального сайта

Установить пакет Jupyter Notebook.

Создал виртуальное окружение проекта:

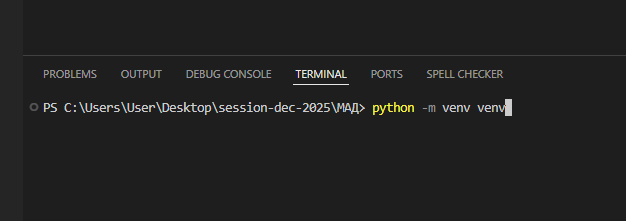


Рисунок 6 – Виртуальное окружение для установки зависимостей

Активировал и поставил зависимости Jupyter и Pandas:

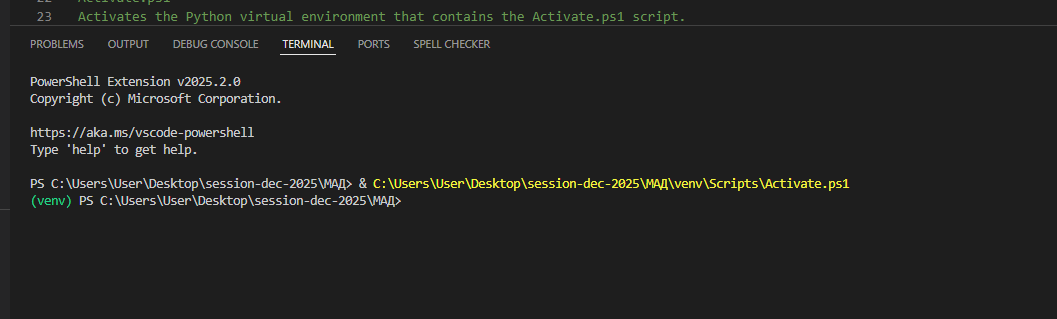


Рисунок 7 – Активация окружения

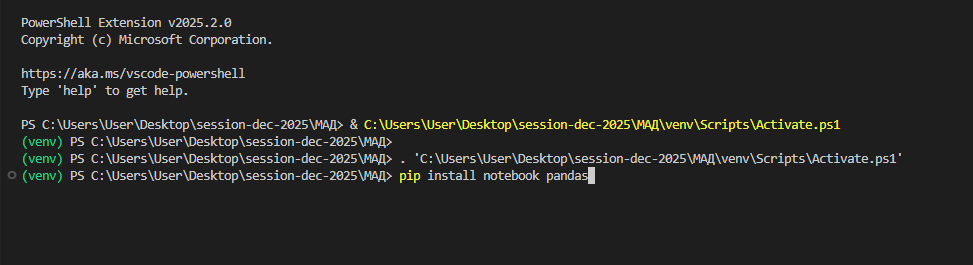


Рисунок 8 – Команда установки



Рисунок 9 – Результат

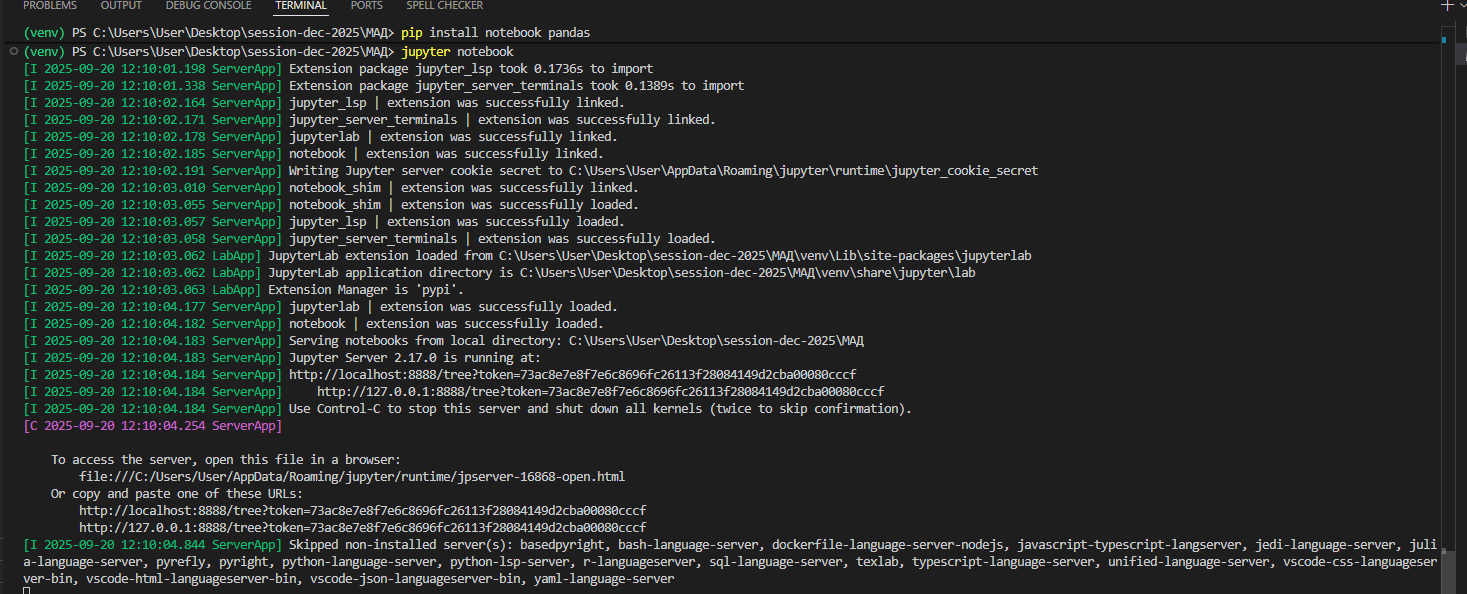


Рисунок 10 – Запуск Jupyter

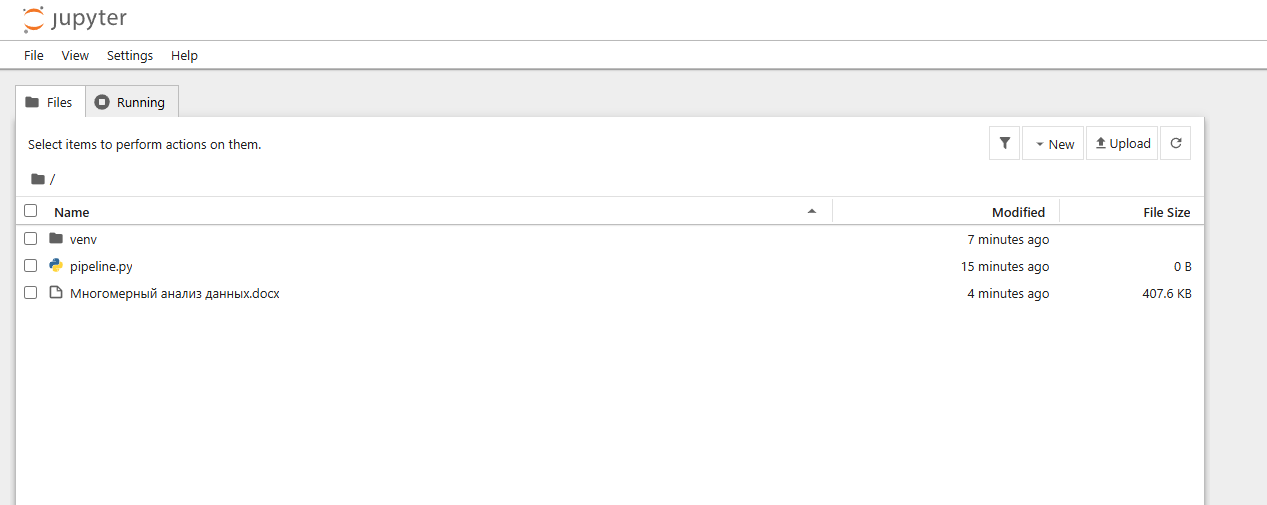


Рисунок 11 – Результат

Ознакомиться (по материалам Интернета) с системой обработки данных в системе Python Pandas. Использовать для этого следующие источники:

1. https://pythonist.ru/pandas-tutorial/
2. https://www.geeksforgeeks.org/how-to-install-python-pandas-on-windows-and-linux/
3. https://www.geeksforgeeks.org/how-to-use-jupyter-notebook-an-ultimate-guide/?ref=lbp
4. https://practicum.yandex.ru/blog/chto-takoe-jupyter-notebook/

В отчёте отметить назначение этих пакетов и их возможности

Я ознакомился с информацией и сделал следующие выводы:

**Pandas:**

- это одна из самых популярных библиотек Python для работы с табличными данными (как в Excel).

**Она используется для:**

1. анализа и обработки данных,
2. очистки и трансформации таблиц,
3. работы с временными рядами,
4. экспорта/импорта данных в разные форматы (CSV, Excel, SQL и т.д.).

**Основные структуры данных в pandas:**

Series — одномерный массив (как столбец таблицы).

DataFrame — двумерная таблица (как Excel-таблица).

**Основные функции для работы с Pandas:**

|  |  |
| --- | --- |
| **Функция / Метод** | **Что делает** |
| pd.read\_csv("file.csv") | Загружает данные из CSV |
| pd.read\_excel("file.xlsx") | Загружает данные из Excel |
| df.head(n) | Первые n строк таблицы |
| df.tail(n) | Последние n строк таблицы |
| df.info() | Информация о столбцах и типах данных |
| df.describe() | Основные статистики (среднее, медиана, min, max) |
| df.shape | Размер таблицы (строки, столбцы) |
| df["col"] | Доступ к столбцу |
| df.loc[5] | Доступ к строке по индексу |
| df.iloc[0:5] | Доступ к строкам по позициям |
| df["new"] = ... | Создать новый столбец |
| df.rename(columns={"old":"new"}) | Переименовать столбцы |
| df.drop("col", axis=1) | Удалить столбец |
| df.dropna() | Удалить строки с пропусками |
| df.fillna(0) | Заменить пропуски на 0 |
| df[df["col"] > 10] | Фильтрация по условию |
| df.groupby("col").agg({...}) | Несколько агрегатов по группам |
| pd.concat([df1, df2]) | Объединить таблицы по строкам |
| pd.merge(df1, df2, on="id") | Объединить по ключу (JOIN) |

Jupyter Notebook — это интерактивная среда для Python (и не только), где код, текст, графики и результаты можно писать и запускать прямо в браузере.

**Используется для:**

1. анализа данных,
2. написания и тестирования кода,
3. создания отчётов и визуализаций,
4. обучения и экспериментов с ML/AI.

Файл Jupyter имеет расширение .ipynb (IPython Notebook).

Основной функционал:

|  |  |
| --- | --- |
| **Функция / Приём** | **Что делает** |
| Shift + Enter | Выполнить текущую ячейку и перейти к следующей |
| Ctrl + Enter | Выполнить текущую ячейку и остаться в ней |
| Alt + Enter | Выполнить ячейку и вставить новую ниже |
| Esc | Выйти из режима редактирования (Command Mode) |
| Enter | Войти в режим редактирования ячейки |
| A (в Command Mode) | Добавить ячейку выше |
| B (в Command Mode) | Добавить ячейку ниже |
| D, D (дважды D) | Удалить ячейку |
| M | Превратить ячейку в Markdown (текст, формулы) |
| Y | Превратить ячейку в Code (Python-код) |
| L | Показать/скрыть номера строк в ячейке |
| Shift + M | Объединить выделенные ячейки |
| 0, 0 (дважды 0) | Перезапустить ядро (kernel) |
| !команда | Выполнить системную команду (например, !ls, !pip install) |
| ?функция | Справка по функции |
| ??функция | Расширенная справка (часто показывает исходный код) |

4.2 Знакомство с Нижегородстатом

Войти на сайт территориального органа Федеральной службы государственной статистики по Нижегородской области (Нижегородстат) - https://52.rosstat.gov.ru/about.

Ознакомиться с материалами и форматом их представления.

Я зашел на сайт, ознакомился с ним.

Сайт имеет раздел «Официальная статистика», где публикуются статистические данные Нижегородской области.

**Форматы документов**

PDF-файлы — отчёты, бюллетени и статистические сборники публикуются в PDF.

Существуют и документы в формате DOCX (например, «Доклад.docx») в некоторых связанных ресурсах, хотя не всегда прямо на Росстат-сайте.

**Тематика материалов**

В статистических публикациях представлены данные по разным областям: демография, социальная сфера, экономика, производство, сельское хозяйство и др.

**Навигация / Структура сайта**

У разделов статистики есть подразделы, где можно выбрать интересующую область или тему статистического исследования.

Часто публикуются временные ряды, сравнения по годам, графики/таблицы.

**Вывод**

Формат презентации: преимущественно PDF, иногда DOCX; таблицы / сводки; ежегодники.

Содержание: охватывает широкие темы — экономика, население, социальная сфера и др.

4.3 Реализовать с помощью Pandas (по материалам лекции) преобразование произвольного текстового файла в формат CSV, а затем в Excel (<https://www.geeksforgeeks.org/convert-csv-to-excel-using-pandas-in-python/>).

Я создал из таблице, приведенной в моем отчете csv и excel файлы с помощью Pandas.

Ход выполнения:

Произвольный текст в директории проекта в файле text.txt:

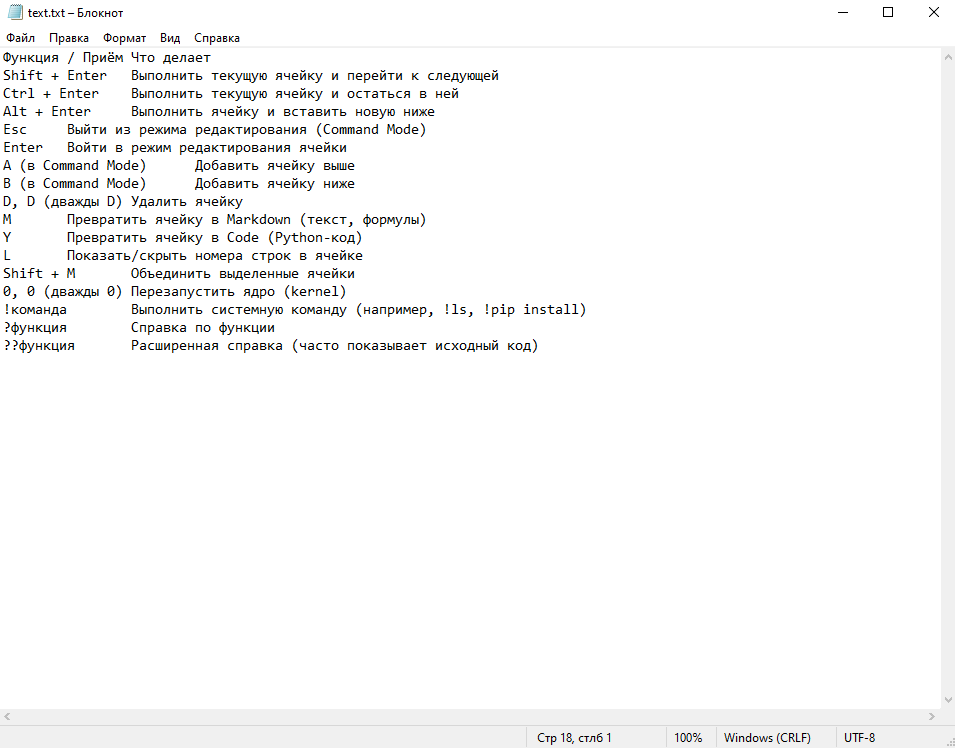


Рисунок 1 – Текст

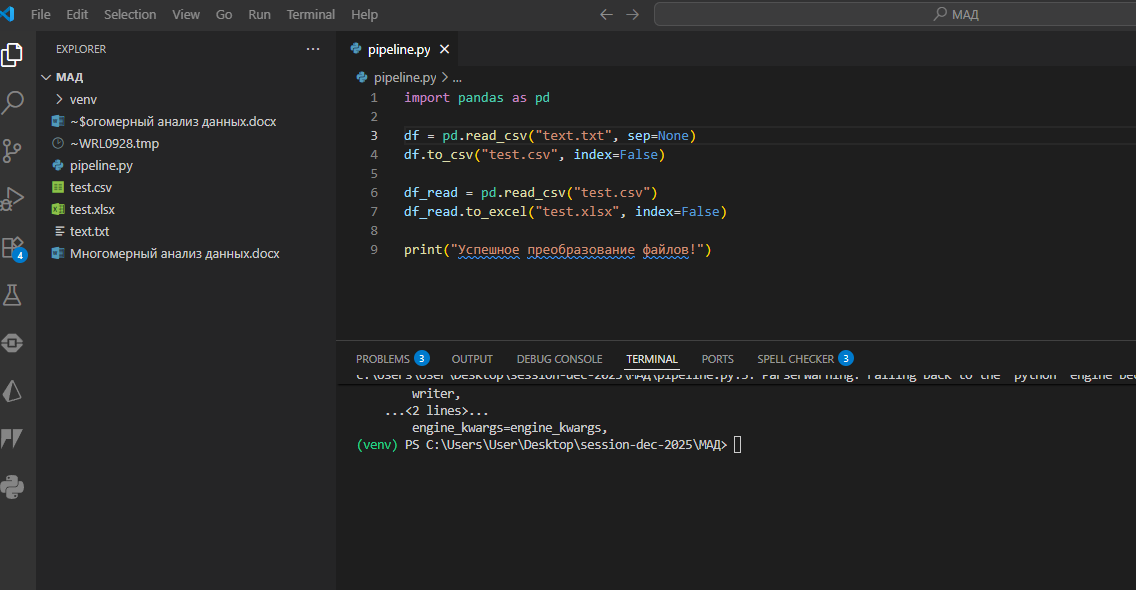


Рисунок 12 – Код

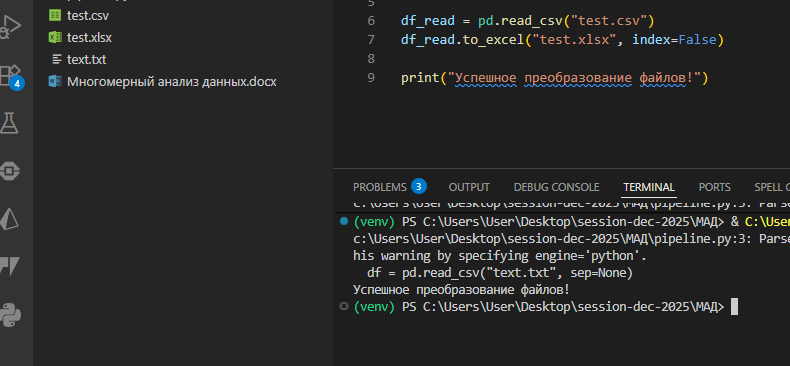


Рисунок 13 - Результат

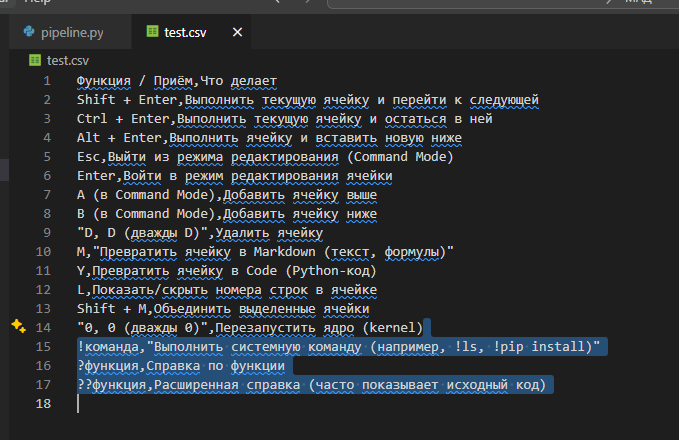


Рисунок 14 – CSV

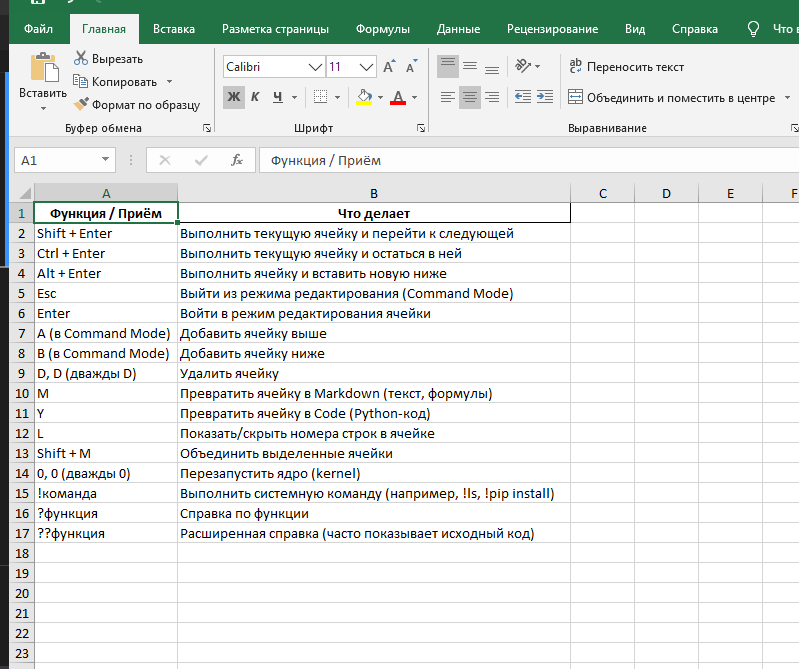


Рисунок 15 – Excel

**Выводы**

В ходе проделанной работы удалось подробно ознакомиться с библиотекой pandas, которая является одним из ключевых инструментов анализа данных в языке Python. Она предоставляет удобные структуры данных и методы, позволяющие быстро обрабатывать таблицы, фильтровать и сортировать строки, выполнять группировки и агрегировать информацию. Использование pandas позволяет значительно упростить работу с большими массивами данных и сделать процесс анализа более наглядным и гибким. Кроме того, было рассмотрено сохранение и загрузка данных из различных форматов, что делает библиотеку универсальной в применении.

Кроме pandas, значительная часть работы была посвящена среде Jupyter Notebook, которая является стандартом де-факто для анализа данных и машинного обучения. Она позволяет сочетать код, текст, графики и результаты выполнения в одном документе, что делает процесс исследования данных более прозрачным и воспроизводимым. Важной особенностью Jupyter является поддержка Markdown и LaTeX, благодаря чему отчёты и эксперименты можно оформлять в удобном и читаемом виде. Также рассмотрены горячие клавиши и базовые команды, которые позволяют ускорить работу и повысить продуктивность.

В итоге освоение pandas и Jupyter Notebook даёт прочную основу для работы с данными в Python. Эти инструменты позволяют не только быстро и эффективно обрабатывать большие объёмы информации, но и документировать процесс анализа в удобной форме. Полученные знания можно применять в учебных и профессиональных проектах, связанных с анализом данных, статистикой или машинным обучением. Таким образом, проделанная работа имеет практическую значимость и открывает возможности для дальнейшего углубленного изучения инструментов анализа данных.