Министерство образования Республики Беларусь
Учреждение образования
«Брестский Государственный технический университет»
Кафедра ИИТ

Лабораторная работа №1
По дисциплине : “ОМО**”**
Тема: “Знакомство с анализом данных
предварительная обработка и визуализация”

Выполнил:
Студент 3 курса
Группы АС-66
Ярома А. И .
Проверил:
Крощенко А.А

Брест 2025

Цель: Получить практические навыки работы с данными с использованием библиотек Pandas для манипуляции и Matplotlib для визуализации. Научиться выполнять основные шаги предварительной обработки данных, такие как очистка, нормализация и работа с различными типами признаков.

Вариант 2

Выборка Boston Housing. Содержит информацию о жилье в разных районах Бостона, включая уровень преступности, количество комнат и медианную стоимость.

Задачи:

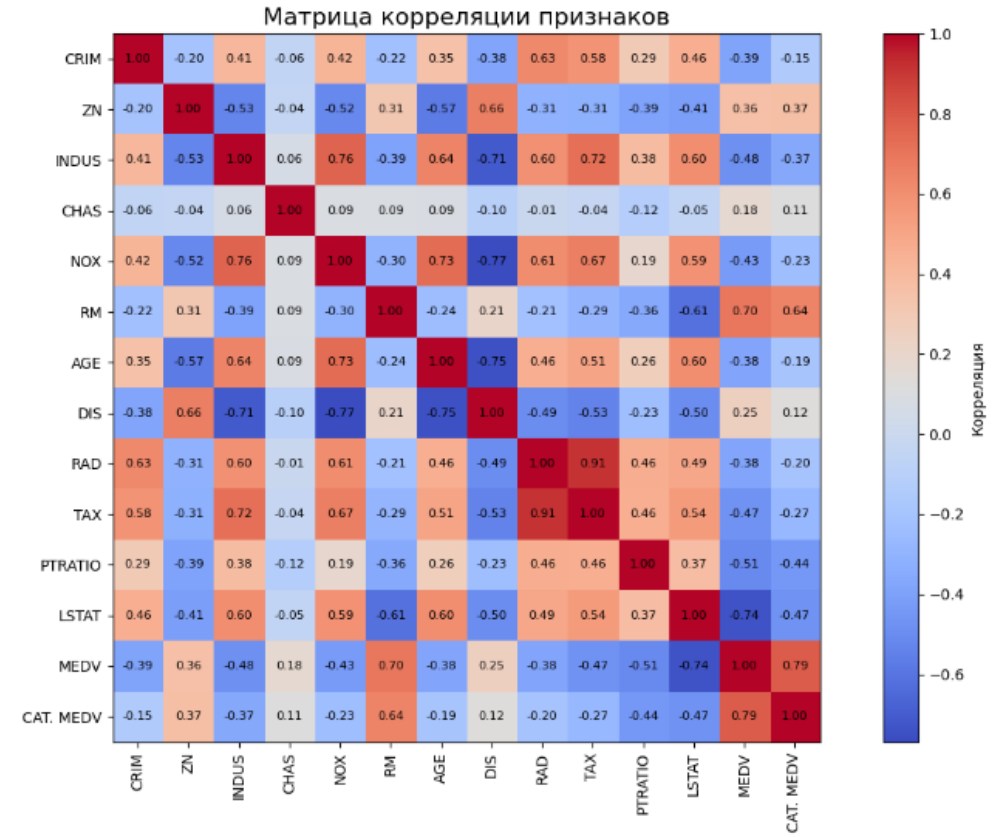
1. Загрузите данные и выведите их основные статистические характеристики (.describe()).
2. Постройте матрицу корреляции и визуализируйте ее с помощью тепловой карты (heatmap).
3. Найдите признак, наиболее сильно коррелирующий с целевой переменной MEDV (медианная стоимость дома).
4. Постройте диаграмму рассеяния (scatter plot) для этого признака и MEDV.
5. Нормализуйте все числовые признаки, приведя их к диапазону от 0 до 1.
6. Визуализируйте распределение уровня преступности (CRIM) с помощью гистограммы.

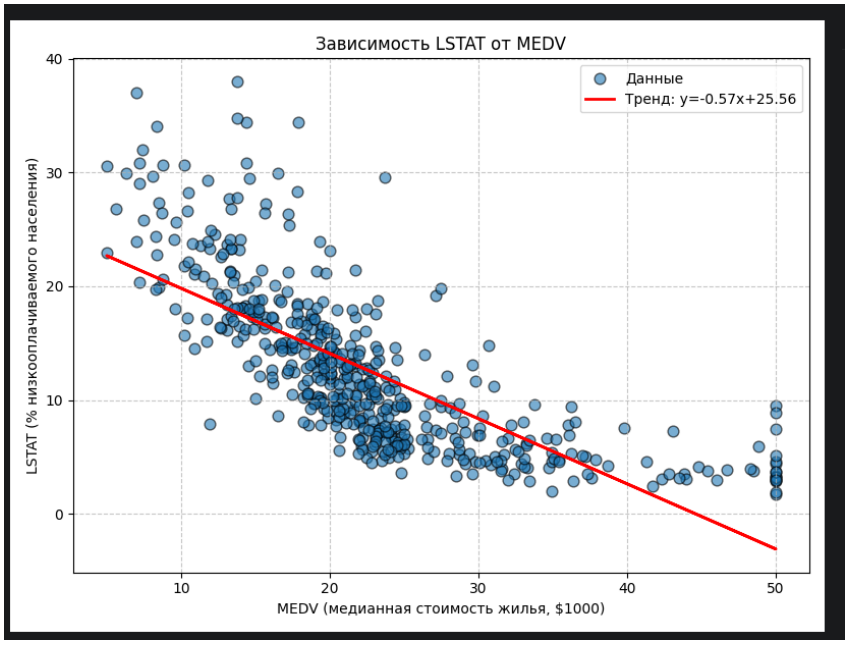
Ход работы

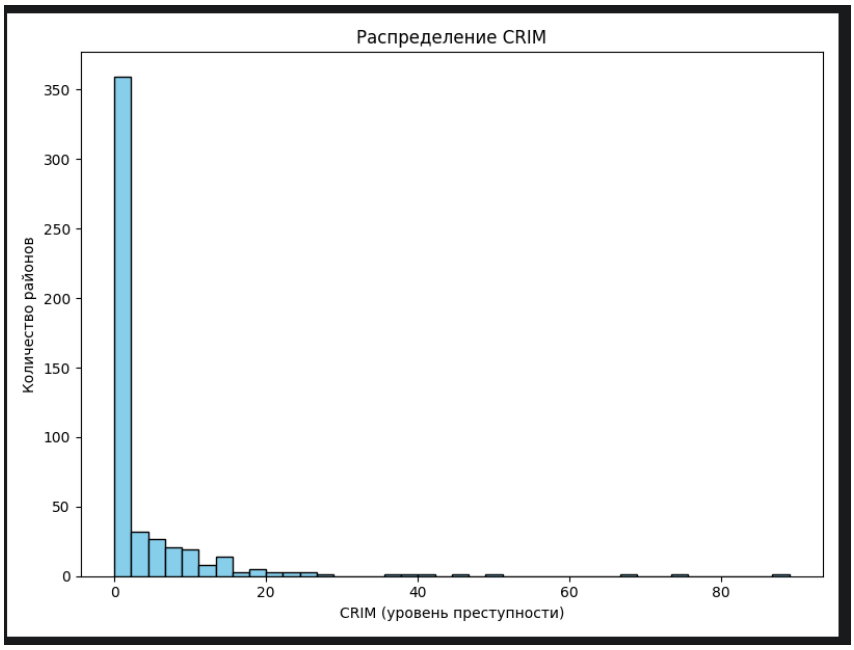
```
import pandas as pnd
import numpy as npy
import matplotlib.pyplot as mpl

def load_dataset(path_to_csv: str) -> pnd.DataFrame:
    return pnd.read_csv(path_to_csv)

def summarize_df(df: pnd.DataFrame) -> None:
    print(">>> Загруженные данные (размер):", df.shape)
    print("\n" + "="*40)
    print("ОБЩАЯ СТАТИСТИКА:")
    print("="*40)
    print(df.describe())
    print("="*40 + "\n")

def plot_correlation_matrix(df: pnd.DataFrame, out_filename: str = "correlation_map.png") -> None:
    print("Построение матрицы корреляций...")
    corr_mtx = df.corr()

    mpl.figure(figsize=(12, 8))
    mpl.imshow(corr_mtx, cmap="coolwarm", interpolation="nearest")
    mpl.colorbar(label="Коэффициент корреляции")
    mpl.xticks(range(len(corr_mtx.columns)), corr_mtx.columns, rotation=90)
    mpl.yticks(range(len(corr_mtx.columns)), corr_mtx.columns)
    mpl.title("Матрица корреляции признаков", fontsize=16)

    for ii in range(len(corr_mtx.columns)):
```

```
        for jj in range(len(corr_mtx.columns)):
            mpl.text(jj, ii, f"{corr_mtx.iloc[ii, jj]:.2f}",
                     ha="center", va="center", color="black", fontsize=8)

    mpl.tight_layout()
    mpl.savefig(out_filename, dpi=300)
    mpl.show()

def plot_medv_vs_lstat(df: pnd.DataFrame, medv_colname: str = "MEDV", lstat_colname: str =
"LSTAT",
                       out_filename: str = "scatter_MEDV_LSTAT_new.png") -> None:
    print("Построение диаграммы разброса MEDV vs LSTAT...")
    medv_series = df[medv_colname]
    lstat_series = df[lstat_colname]

    mpl.figure(figsize=(8, 6))
    mpl.scatter(medv_series, lstat_series, alpha=0.6, edgecolors="k", s=60, label="Набор данных")

    slope, intercept = npy.polyfit(medv_series, lstat_series, 1)
    mpl.plot(medv_series, slope * medv_series + intercept, linewidth=2, label=f"Тренд:
y={slope:.2f}x+{intercept:.2f}")

    mpl.xlabel("MEDV (медианная стоимость, $1000)")
    mpl.ylabel("LSTAT (% низкооплачиваемых)")
    mpl.title("Зависимость LSTAT от MEDV")
    mpl.legend()
    mpl.grid(True, linestyle="--", alpha=0.7)
    mpl.tight_layout()
    mpl.savefig(out_filename, dpi=300)
    mpl.show()

def plot_crim_histogram(df: pnd.DataFrame, crim_colname: str = "CRIM", out_filename: str =
"hist_CRIM_new.png") -> None:
    print("Построение гистограммы распределения CRIM...")
    mpl.figure(figsize=(8, 6))
    mpl.hist(df[crim_colname], bins=40, edgecolor="black")
    mpl.xlabel("CRIM (уровень преступности)")
    mpl.ylabel("Число районов")
    mpl.title("Распределение CRIM")
    mpl.tight_layout()
    mpl.savefig(out_filename, dpi=300)
    mpl.show()

def minmax_normalize(df: pnd.DataFrame) -> pnd.DataFrame:
    print("Выполняю нормализацию (min-max)...")
    return (df - df.min()) / (df.max() - df.min())

def main():
    print("Загрузка набора данных...")
    dataset = load_dataset("BostonHousing.csv")
    print("Данные загружены успешно.")
    summarize_df(dataset)

    plot_correlation_matrix(dataset, out_filename="corr_matrix_renamed.png")
    plot_medv_vs_lstat(dataset, medv_colname="MEDV", lstat_colname="LSTAT",
```

```
out_filename="scatter_MEDV_LSTAT_renamed.png")
    plot_crim_histogram(dataset, crim_colname="CRIM", out_filename="hist_CRIM_renamed.png")

    normalized_dataset = minmax_normalize(dataset)
    print("\nПервые строки нормализованных признаков:")
    print(normalized_dataset.head())

    print("\nГотово — все графики сохранены и показаны.")

if __name__ == "__main__":
    main()
    input("Нажмите Enter, чтобы завершить программу...")
```








```
Построение матрицы корреляций...
Построение диаграммы разброса MEDV vs LSTAT...
Построение гистограммы распределения CRIM...
Выполняю нормализацию (min-max)...

Первые строки нормализованных признаков:
       CRIM    ZN     INDUS  CHAS  ...    PTRATIO     LSTAT      MEDV  CAT. MEDV
0  0.000000  0.18  0.067815   0.0  ...   0.287234  0.089680  0.422222        0.0
1  0.000236  0.00  0.242302   0.0  ...   0.553191  0.204470  0.368889        0.0
2  0.000236  0.00  0.242302   0.0  ...   0.553191  0.063466  0.660000        1.0
3  0.000293  0.00  0.063050   0.0  ...   0.648936  0.033389  0.631111        1.0
4  0.000705  0.00  0.063050   0.0  ...   0.648936  0.099338  0.693333        1.0

[5 rows x 14 columns]

Готово — все графики сохранены и показаны.
```

Вывод: Получили практические навыки работы с данными с использованием библиотек Pandas для манипуляции и Matplotlib для визуализации. Научились выполнять основные шаги предварительной обработки данных, такие как очистка, нормализация и работа с различными типами признаков.