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

Лабораторная работа №1
По дисциплине «ОМО»

Выполнил:
Студент 3-го курса
Группы АС-65
Грущинский Д.Д.
Проверил:
Крощенко А.А.

Брест 2025

Цель работы: 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.

Ход работы

Общее задание:

1. Загрузить предложенный набор данных (по вариантам) в DataFrame библиотеки Pandas.
2. Провести исследовательский анализ: изучить типы данных, количество пропусков, основные статистические показатели (среднее, медиана, стандартное отклонение).
3. Обработать пропущенные значения (например, заполнить средним значением или удалить строки/столбцы).
4. Преобразовать категориальные признаки в числовые с помощью метода One-Hot Encoding.
5. Выполнить нормализацию или стандартизацию числовых признаков.
6. Построить несколько графиков для визуализации данных (гистограммы, диаграммы рассеяния) и сделать выводы о зависимостях между признаками.
7. Написать отчет, создать пул-реквест в репозиторий с кодом решения и отчетом в формате pdf.

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.

Код программы:

```
from pandas import *
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
import numpy as np
class BostonHousingAnalyze:
    def __init__(self, dataset_path: str = "BostonHousing.csv"):
        self.boston = read_csv(dataset_path)
        self.correlation_matrix = self.boston.corr()
        self.medv_correlations =
self.correlation_matrix['MEDV'].abs().sort_values(ascending=False)
        self.most_correlated_feature = self.medv_correlations.index[1]
        self.most_correlated_value = self.medv_correlations[1]
        self.scaler = MinMaxScaler()
        self.numeric_columns =
self.boston.select_dtypes(include=[np.number]).columns
        self.boston_normalized = self.boston.copy()
        self.boston_normalized[self.numeric_columns] =
self.scaler.fit_transform(self.boston[self.numeric_columns])
    @staticmethod
    def beautify(func):
        def wrapper(*args, **kwargs):
            print(f"\nВыполнение: {func.__name__}\n"+"="*80+"\n")
            result = func(*args, **kwargs)
            print("\n"+"="*80+"\n")
            return result
        return wrapper
    @beautify
    def zad_1(self):
        print(f"Основные статистические
характеристики:\n{self.boston.describe()}")
    @beautify
    def zad_2(self):
        plt.figure(figsize=(12, 10))
        sns.heatmap(self.correlation_matrix, annot=True,
cmap='coolwarm', center=0, fmt='.2f')
        plt.title('Матрица корреляции Boston Housing')
```

```
            plt.tight_layout()

            plt.show()

        @beautify

        def zad_3(self):

            print(f"Признак, наиболее сильно коррелирующий с MEDV:
{self.most_correlated_feature}")

            print(f"Коэффициент корреляции:
{self.most_correlated_value:.3f}")

            print(f"Корреляции всех признаков с
MEDV:\n{self.medv_correlations}")

        @beautify

        def zad_4(self):

            plt.figure(figsize=(10, 6))

            plt.scatter(self.boston[self.most_correlated_feature],
self.boston['MEDV'], alpha=0.6)

            plt.xlabel(self.most_correlated_feature.upper())

            plt.ylabel('MEDV (Медианная стоимость)')

            plt.title(f'Диаграмма рассеяния:
{self.most_correlated_feature.upper()} vs MEDV')

            plt.grid(True, alpha=0.3)

            plt.tight_layout()

            plt.show()

        @beautify

        def zad_5(self):

            print("Данные после нормализации (первые 5 строк):")

            print(self.boston_normalized.head())

            print("\nПроверка диапазона (min/max) после нормализации:")

            print(self.boston_normalized[self.numeric_columns].agg(['min',
'max']))

        @beautify

        def zad_6(self):

            plt.figure(figsize=(12, 5))

            plt.subplot(1, 2, 1)

            plt.hist(self.boston['CRIM'], bins=30, alpha=0.7,
color='skyblue', edgecolor='black')

            plt.xlabel('Уровень преступности (CRIM)')

            plt.ylabel('Частота')

            plt.title('Распределение CRIM (оригинальные данные)')

            plt.grid(True, alpha=0.3)
```

```
            plt.subplot(1, 2, 2)
            plt.hist(self.boston_normalized['CRIM'], bins=30, alpha=0.7,
color='lightcoral', edgecolor='black')
            plt.xlabel('Уровень преступности (CRIM) нормализованный')
            plt.ylabel('Частота')
            plt.title('Распределение CRIM (после нормализации)')
            plt.grid(True, alpha=0.3)
            plt.tight_layout()
            plt.show()
        def __call__(self):
            self.zad_1()
            self.zad_2()
            self.zad_3()
            self.zad_4()
            self.zad_5()
            self.zad_6()
    if __name__ == "__main__":
        BostonHousingAnalyze()()
```

В данном коде изложены все задания данной лабораторной работы (в методах заданий). Разберем результаты вывода данной программы.

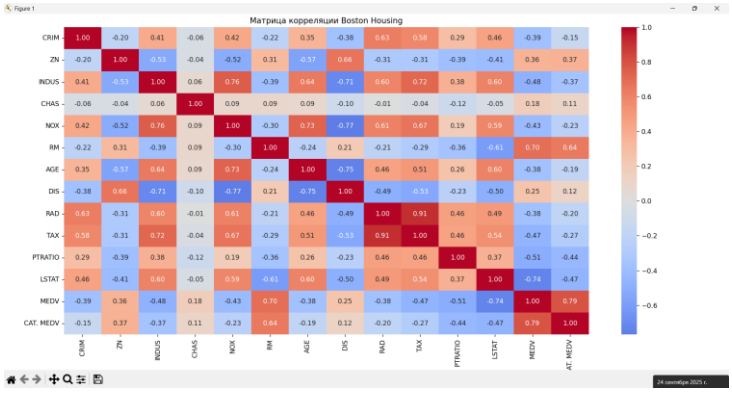
Вывод результатов первого задания:

```
Выполнение: zad_1
============================================================

Основные статистические характеристики:
             CRIM          ZN       INDUS        CHAS         NOX          RM  ...         RAD         TAX     PTRATIO       LSTAT        MEDV    CAT. MEDV
count  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000  ...  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000
mean     3.613524   11.363636   11.136779    0.069170    0.554695    6.284634  ...    9.549407  408.237154   18.455534   12.653063   22.532806    0.166008
std      8.601545   23.322453    6.860353    0.253994    0.115878    0.702617  ...    8.707259  168.537116    2.164946    7.141062    9.197104    0.372456
min      0.006320    0.000000    0.460000    0.000000    0.385000    3.561000  ...    1.000000  187.000000   12.600000    1.730000    5.000000    0.000000
25%      0.082045    0.000000    5.190000    0.000000    0.449000    5.885500  ...    4.000000  279.000000   17.400000    6.950000   17.025000    0.000000
50%      0.256510    0.000000    9.690000    0.000000    0.538000    6.208500  ...    5.000000  330.000000   19.050000   11.360000   21.200000    0.000000
75%      3.677083   12.500000   18.100000    0.000000    0.624000    6.623500  ...   24.000000  666.000000   20.200000   16.955000   25.000000    0.000000
max     88.976200  100.000000   27.740000    1.000000    0.871000    8.780000  ...   24.000000  711.000000   22.000000   37.970000   50.000000    1.000000

[8 rows x 14 columns]
```

Вывод результатов второго задания:



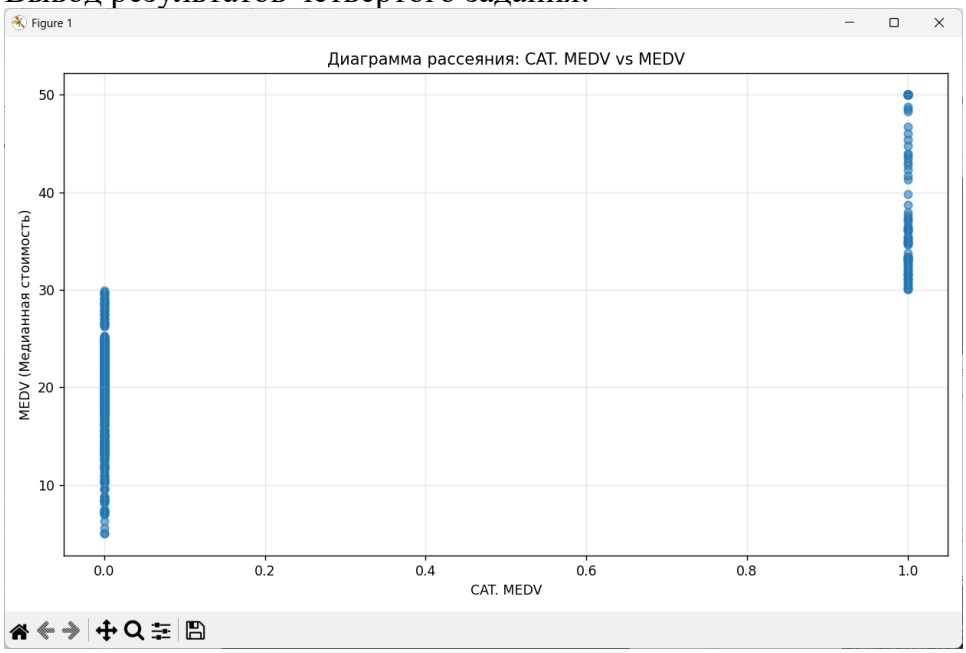
Вывод результатов третьего задания:

```
Выполнение: zad_3
================================================================================

Признак, наиболее сильно коррелирующий с MEDV: CAT. MEDV
Коэффициент корреляции: 0.790
Корреляции всех признаков с MEDV:
MEDV         1.000000
CAT. MEDV    0.789789
LSTAT        0.737663
RM           0.695360
PTRATIO      0.507787
INDUS        0.483725
TAX          0.468536
NOX          0.427321
CRIM         0.388305
RAD          0.381626
AGE          0.376955
ZN           0.360445
DIS          0.249929
CHAS         0.175260
Name: MEDV, dtype: float64

================================================================================
```

Вывод результатов четвёртого задания:



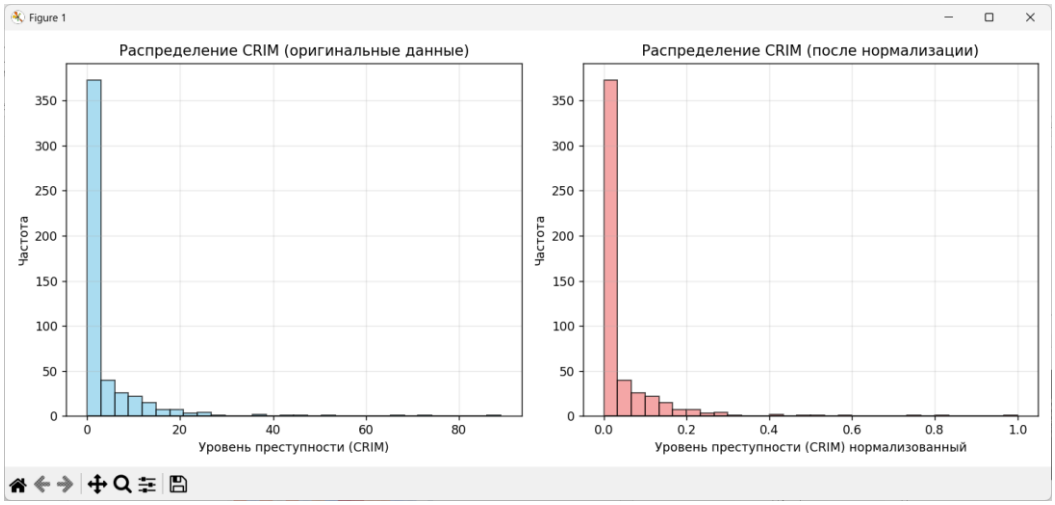
Вывод результатов пятого задания:

```
Выполнение: zad_5
======================================================================

Данные после нормализации (первые 5 строк):
       CRIM    ZN     INDUS  CHAS       NOX        RM       AGE       DIS       RAD       TAX   PTRATIO     LSTAT      MEDV  CAT. M
0  0.000000  0.18  0.067815   0.0  0.314815  0.577505  0.641607  0.269203  0.000000  0.208015  0.287234  0.089680  0.422222
1  0.000236  0.00  0.242302   0.0  0.172840  0.547998  0.782698  0.348962  0.043478  0.104962  0.553191  0.204470  0.368889
2  0.000236  0.00  0.242302   0.0  0.172840  0.694386  0.599382  0.348962  0.043478  0.104962  0.553191  0.063466  0.660000
3  0.000293  0.00  0.063050   0.0  0.150206  0.658555  0.441813  0.448545  0.086957  0.066794  0.648936  0.033389  0.631111
4  0.000705  0.00  0.063050   0.0  0.150206  0.687105  0.528321  0.448545  0.086957  0.066794  0.648936  0.099338  0.693333

Проверка диапазона (min/max) после нормализации:
     CRIM   ZN  INDUS  CHAS  NOX   RM  AGE  DIS  RAD  TAX  PTRATIO  LSTAT  MEDV  CAT. MEDV
min   0.0  0.0    0.0   0.0  0.0  0.0  0.0  0.0  0.0  0.0      0.0    0.0   0.0        0.0
max   1.0  1.0    1.0   1.0  1.0  1.0  1.0  1.0  1.0  1.0      1.0    1.0   1.0        1.0
```

Вывод результатов шестого задания:



Вывод: мы приобрели практические знания по работе с Pandas, Matplotlib, а также научились анализировать датасеты для дальнейшего обучения моделей на их основе.