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-66
Ярома А.И
Проверила:
Крощенко А.А

Брест 2025

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:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler, StandardScaler, OneHotEncoder

# === 1. Загрузка данных из локального файла ===
# Если lab1.py и BostonHousing.csv находятся в одной папке:
df = pd.read_csv("BostonHousing.csv")

print("Первые 5 строк данных:")
print(df.head())

# === 2. Исследовательский анализ данных ===
print("\nИнформация о данных:")
print(df.info())

print("\nКоличество пропусков по столбцам:")
print(df.isnull().sum())

print("\nОсновные статистические показатели (.describe()):")
print(df.describe())

print("\nДополнительные показатели:")
print(f"Среднее значение MEDV: {df['MEDV'].mean():.2f}")
```

```
print(f"Медиана MEDV: {df['MEDV'].median():.2f}")
print(f"Стандартное отклонение MEDV: {df['MEDV'].std():.2f}")

# === 3. Обработка пропусков ===
if df.isnull().sum().sum() > 0:
    df = df.fillna(df.mean(numeric_only=True))
    print("\nПропуски были заполнены средними значениями.")
else:
    print("\nПропусков не обнаружено — обработка не требуется.")

# === 4. One-Hot Encoding для категориальных признаков ===
categorical_cols = df.select_dtypes(include=["object"]).columns.tolist()
if "CHAS" in df.columns and df["CHAS"].nunique() <= 2:
    encoder = OneHotEncoder(drop='if_binary', sparse_output=False)
    encoded = encoder.fit_transform(df[["CHAS"]])
    encoded_df = pd.DataFrame(encoded, columns=['CHAS_1'])
    df = pd.concat([df.drop(columns=["CHAS"]), encoded_df], axis=1)
    print("\nКатегориальный признак CHAS преобразован методом One-Hot Encoding.")

elif len(categorical_cols) > 0:
    df = pd.get_dummies(df, columns=categorical_cols, drop_first=True)
    print(f"\nПреобразованы категориальные признаки: {categorical_cols}")
else:
    print("\nКатегориальных признаков для кодирования не найдено.")

# === 5. Нормализация и стандартизация числовых признаков ===
scaler_minmax = MinMaxScaler()
scaler_std = StandardScaler()

num_cols = df.select_dtypes(include=[np.number]).columns
normalized = pd.DataFrame(scaler_minmax.fit_transform(df[num_cols]), columns=num_cols)
standardized = pd.DataFrame(scaler_std.fit_transform(df[num_cols]), columns=num_cols)

print("\nПример нормализованных данных (0–1):")
print(normalized.head())
print("\nПример стандартизованных данных (Z-score):")
print(standardized.head())

# === 6. Матрица корреляции и тепловая карта ===
corr = df.corr(numeric_only=True)
plt.figure(figsize=(10, 8))
sns.heatmap(corr, annot=True, fmt=".2f", cmap="coolwarm")
plt.title("Матрица корреляции признаков")
plt.show()
```

```
# === 7. Признак, наиболее коррелирующий с MEDV ===
target_corr = corr["MEDV"].drop("MEDV")
strongest_feature = target_corr.abs().idxmax()
print(f"\nНаиболее сильно коррелирующий признак с MEDV: {strongest_feature} (r = {target_corr[strongest_feature]:.3f})")
# === 8. Диаграмма рассеяния MEDV vs этот признак ===
plt.figure(figsize=(6, 4))
sns.scatterplot(x=df[strongest_feature], y=df["MEDV"])
plt.xlabel(strongest_feature)
plt.ylabel("MEDV")
plt.title(f"Зависимость MEDV от {strongest_feature}")
plt.show()
# === 9. Гистограммы ===
plt.figure(figsize=(6, 4))
plt.hist(df["CRIM"], bins=30, edgecolor="black")
plt.title("Распределение уровня преступности (CRIM)")
plt.xlabel("CRIM")
plt.ylabel("Частота")
plt.show()
plt.figure(figsize=(6, 4))
plt.hist(df["MEDV"], bins=30, edgecolor="black", color="orange")
plt.title("Распределение медианной стоимости домов (MEDV)")
plt.xlabel("MEDV")
plt.ylabel("Частота")
plt.show()
# === 10. Дополнительная визуализация: RM vs LSTAT ===
plt.figure(figsize=(6, 4))
sns.scatterplot(x=df["LSTAT"], y=df["RM"])
plt.title("Связь между LSTAT и RM")
plt.xlabel("LSTAT (доля низкого статуса населения)")
plt.ylabel("RM (среднее число комнат)")
plt.show()
# === Итоговые выводы ===
print("\nВыводы:")
print(f"- Признак {strongest_feature} имеет наибольшую корреляцию с MEDV (r = {target_corr[strongest_feature]:.3f}).")
print("- Пропусков в данных не обнаружено.")
print("- Признаки успешно нормализованы и стандартизованы.")
print("- Распределение CRIM скошено в сторону малых значений, MEDV близко к нормальному.")
print("- Чем выше LSTAT, тем ниже MEDV. Чем больше RM — тем выше стоимость жилья.")
```

Результат:

Таблица 1

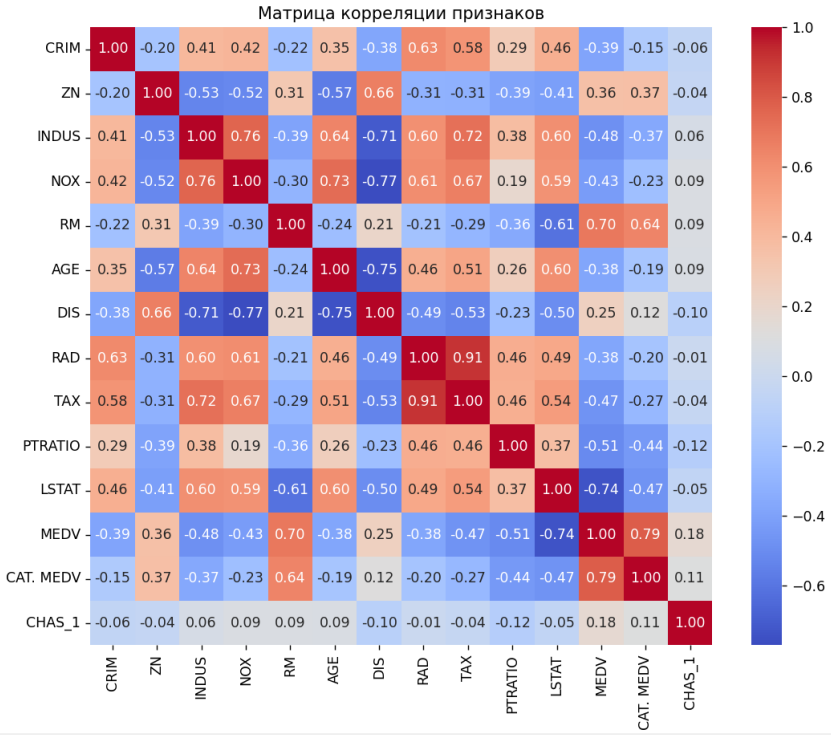


Таблица 2

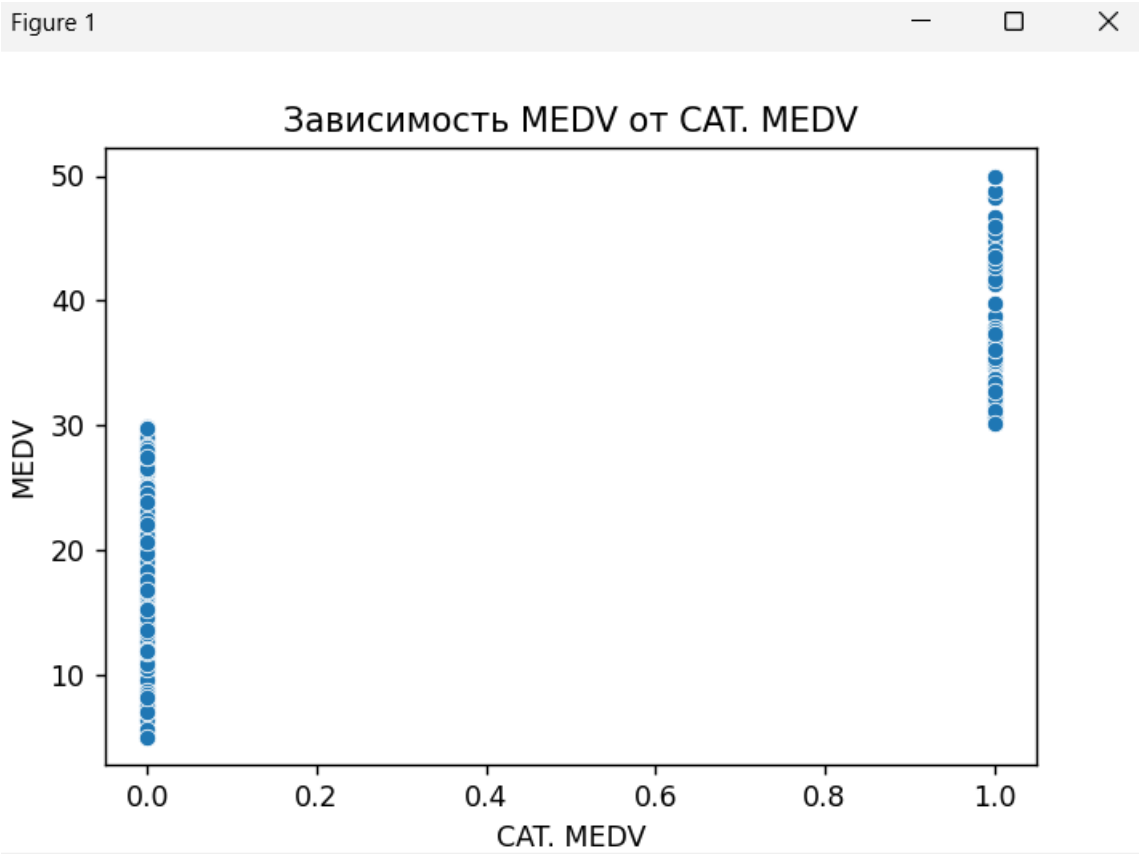


Таблица 3

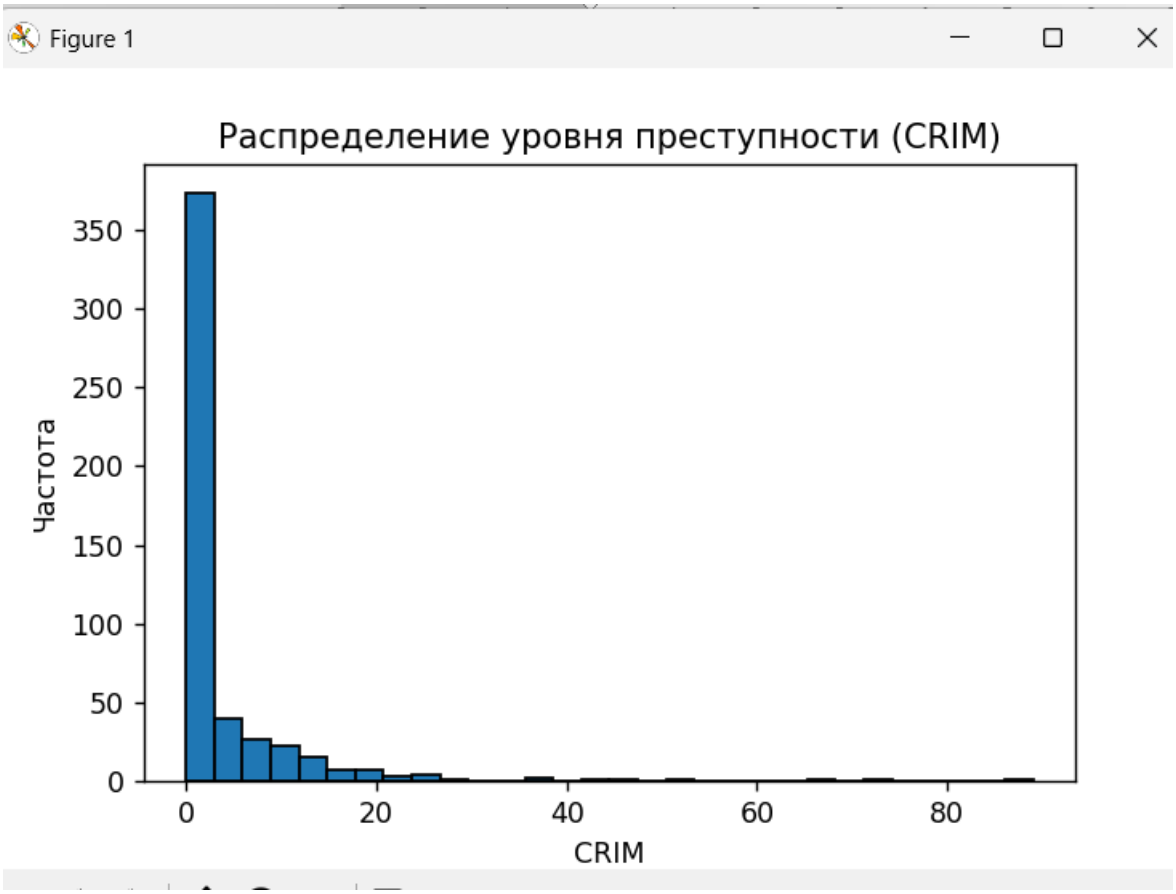


Таблица 4

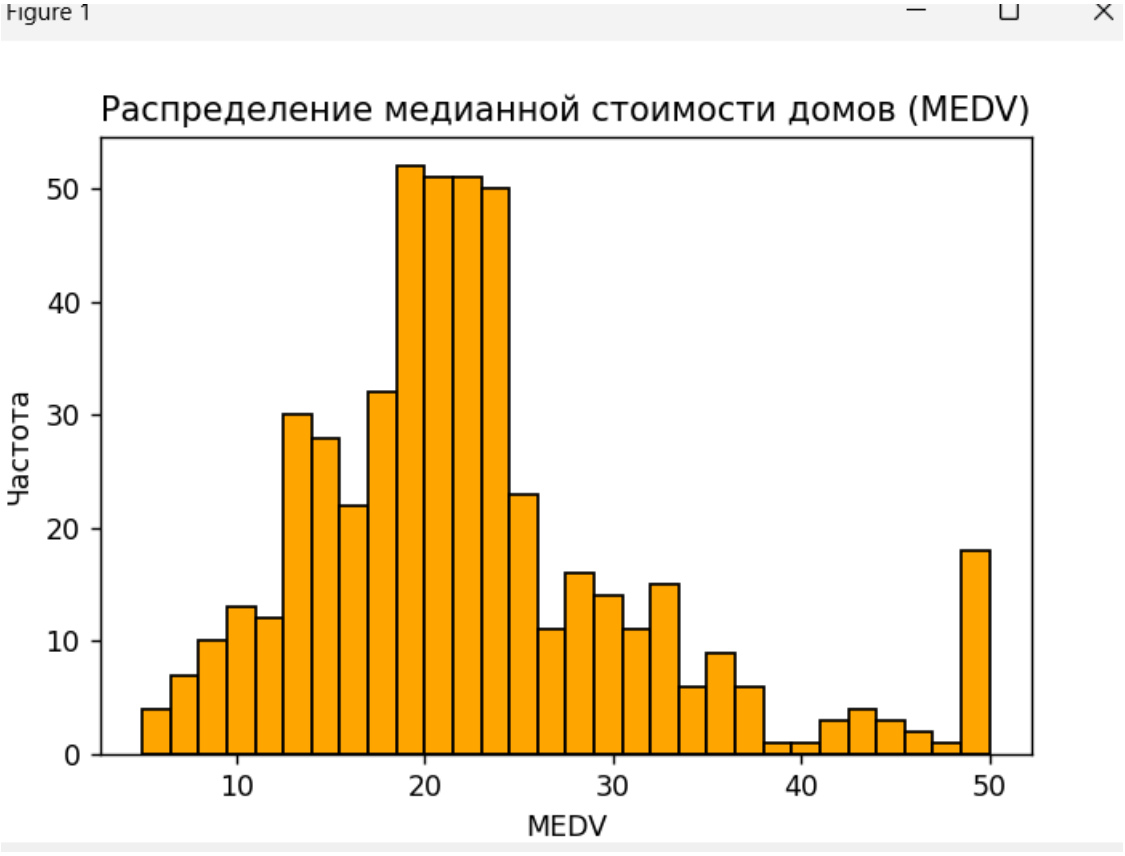
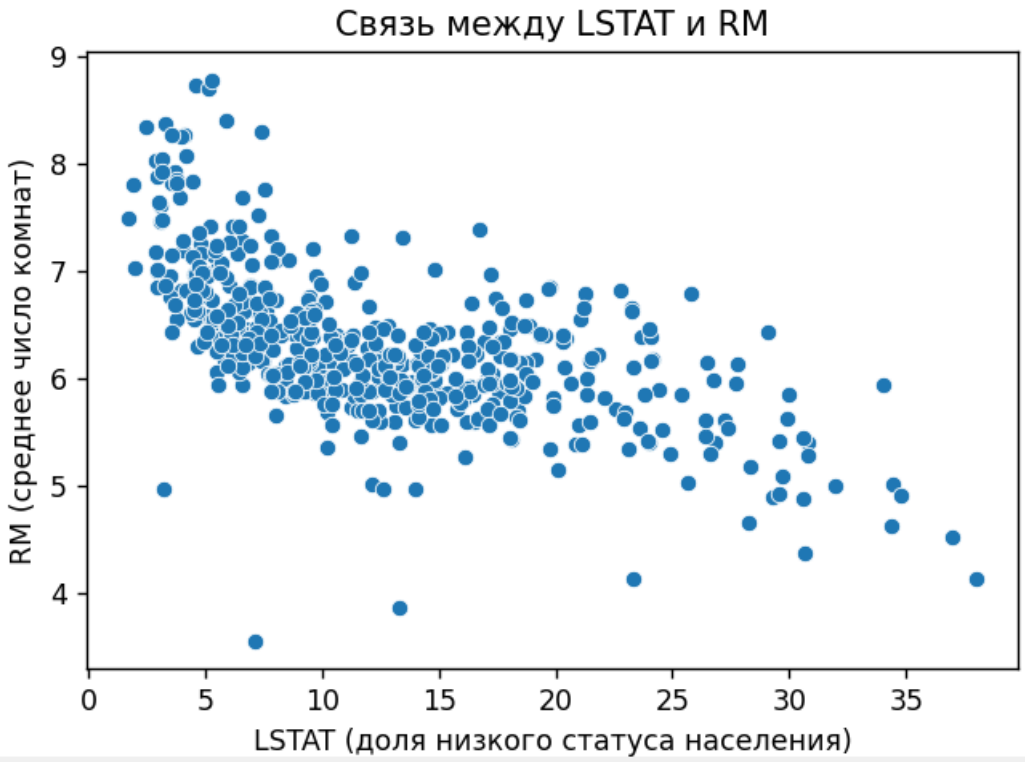


Таблица 5




```
Нормализованные признаки (первые строки):
      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]

Process finished with exit code 0
```