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

Лабораторная работа №1
По дисциплине: «Основы машинного обучения»
Тема**: «Знакомство с анализом данных: предварительная обработка и визуализация»**

Выполнил:
Студент 3-го курса
Группы АС-65
Кисель М. С.
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.**

Используемые инструменты: Python, Pandas, Matplotlib, NumPy, Jupyter Notebook / Google Colab / PyCharm

Вариант 8

Выборка Pima Indians Diabetes. Содержит медицинские показатели женщин из племени Пима и информацию о наличии у них диабета.

Задачи:

1. Загрузите данные и выведите их статистические характеристики.

```python
import os
import pandas as pd
#Заголовки в csv-файле(Задаем вручную)
columns = [
    "Pregnancies",
    "Glucose",
    "BloodPressure",
    "SkinThickness",
    "Insulin",
    "BMI",
    "DiabetesPedigreeFunction",
    "Age",
    "Outcome"
]

# Загрузка и первичный просмотр данных
os.chdir("d:/Универ/ОМО/ОМО2025/Лаба1/")
df = pd.read_csv("pima-indians-diabetes.csv", comment="#", names=columns, header=None)
print("Первые 5 строк:")
print(df.head())
print("\nИнформация о данных:")
print(df.info())
print("\nСтатистика по числовым признакам:")
print(df.describe())
```

```
Первые 5 строк:
   Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  DiabetesPedigreeFunction  Age  Outcome
0            6      148             72             35        0  33.6                     0.627   50        1
1            1       85             66             29        0  26.6                     0.351   31        0
2            8      183             64              0        0  23.3                     0.672   32        1
3            1       89             66             23       94  28.1                     0.167   21        0
4            0      137             40             35      168  43.1                     2.288   33        1

Информация о данных:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
None
```

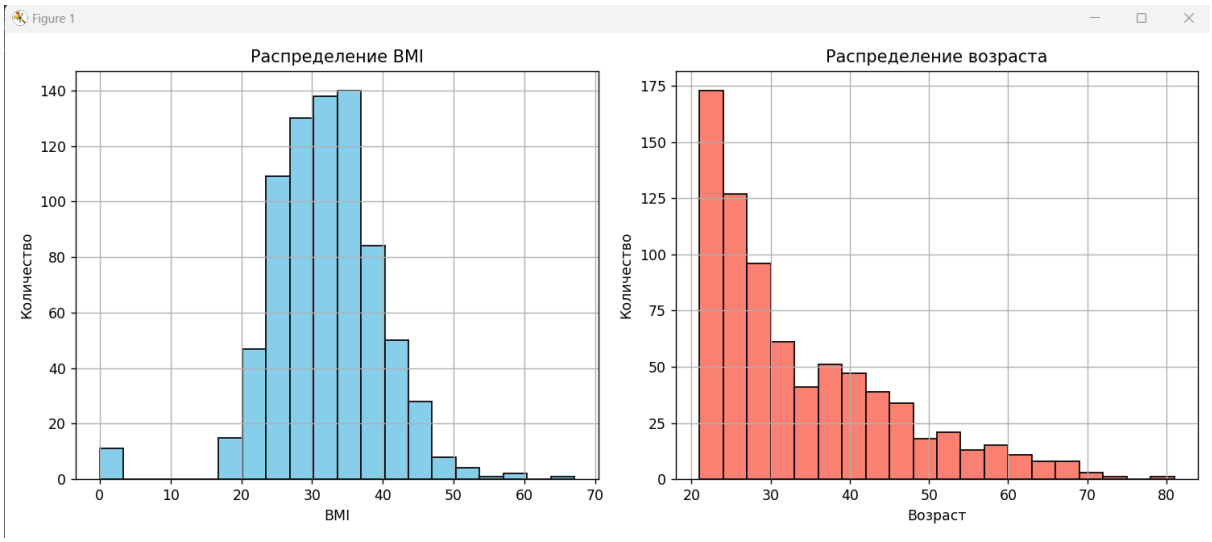
```
Статистика по числовым признакам:
       Pregnancies     Glucose  BloodPressure  SkinThickness  ...         BMI  DiabetesPedigreeFunction         Age     Outcome
count   768.000000  768.000000     768.000000     768.000000  ...  768.000000                768.000000  768.000000  768.000000
mean      3.845052  120.894531      69.105469      20.536458  ...   31.992578                  0.471876   33.240885    0.348958
std       3.369578   31.972618      19.355807      15.952218  ...    7.884160                  0.331329   11.760232    0.476951
min       0.000000    0.000000       0.000000       0.000000  ...    0.000000                  0.078000   21.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000  ...   27.300000                  0.243750   24.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000  ...   32.000000                  0.372500   29.000000    0.000000
75%       6.000000  140.250000      80.000000      32.000000  ...   36.600000                  0.626250   41.000000    1.000000
max      17.000000  199.000000     122.000000      99.000000  ...   67.100000                  2.420000   81.000000    1.000000

[8 rows x 9 columns]
```

2. Проанализируйте столбцы Glucose, BloodPressure, SkinThickness. Нулевые значения в них, скорее всего, являются ошибками. Замените их медианным значением соответствующего столбца.

```python
# Замена скрытых пропусков (нулей) на медиану
cols = ['Glucose', 'BloodPressure', 'SkinThickness']
for c in cols:
    median = df[c].median()
    df[c] = df[c].replace(0, median)
```

3. Постройте гистограммы для признаков BMI и Age.

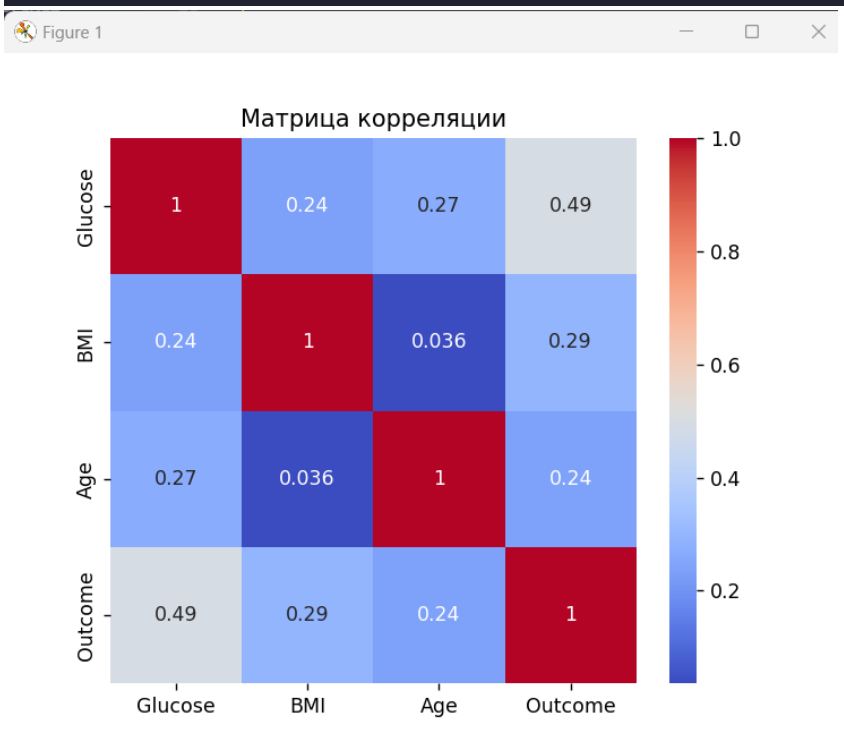
```python
import matplotlib.pyplot as plt
import seaborn as sns

# Визуализация распределений BMI и Age
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
df['BMI'].hist(bins=20, color='skyblue', edgecolor='black')
plt.title('Распределение BMI')
plt.xlabel('BMI')
plt.ylabel('Количество')

plt.subplot(1, 2, 2)
df['Age'].hist(bins=20, color='salmon', edgecolor='black')
plt.title('Распределение возраста')
plt.xlabel('Возраст')
plt.ylabel('Количество')

plt.tight_layout()
plt.show()
```



4. Создайте матрицу корреляции только для признаков Glucose, BMI, Age и Outcome

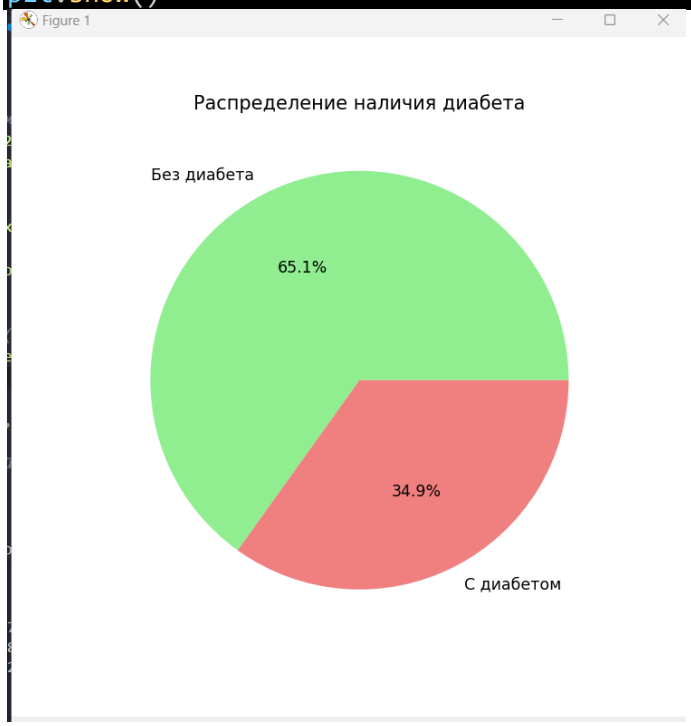
```
# Матрица корреляции (Glucose, BMI, Age, Outcome)
subset = df[['Glucose', 'BMI', 'Age', 'Outcome']]
corr = subset.corr()
print("\nМатрица корреляции:")
print(corr)

plt.figure(figsize=(6, 5))
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Матрица корреляции')
plt.show()
```

```
Матрица корреляции:
            Glucose       BMI       Age   Outcome
Glucose    1.000000  0.235035  0.266909  0.492782
BMI        0.235035  1.000000  0.036242  0.292695
Age        0.266909  0.036242  1.000000  0.238356
Outcome    0.492782  0.292695  0.238356  1.000000
```



5. Визуализируйте распределение Outcome (наличие диабета) с помощью круговой диаграммы.

```
# Круговая диаграмма распределения Outcome
outcome_counts = df['Outcome'].value_counts()
plt.figure(figsize=(6, 6))
plt.pie(outcome_counts, labels=['Без диабета', 'С диабетом'], autopct='%1.1f%%',
colors=['lightgreen', 'lightcoral'])
plt.title('Распределение наличия диабета')
plt.show()
```



6. Примените стандартизацию ко всем признакам, кроме Outcome.

```
from sklearn.preprocessing import StandardScaler

# Стандартизация признаков (кроме Outcome)
scaler = StandardScaler()
features = df.drop('Outcome', axis=1)
scaled_features = scaler.fit_transform(features)
df_scaled = pd.DataFrame(scaled_features, columns=features.columns)
df_scaled['Outcome'] = df['Outcome']

print("\nСтандартизированные данные:")
print(df_scaled.head())
```

```
Стандартизированные данные:
   Pregnancies   Glucose  BloodPressure  SkinThickness   Insulin       BMI  DiabetesPedigreeFunction       Age  Outcome
0     0.639947  0.866045      -0.031990       0.831114 -0.692891  0.204013                  0.468492  1.425995        1
1    -0.844885 -1.205066      -0.528319       0.180566 -0.692891 -0.684422                 -0.365061 -0.190672        0
2     1.233880  2.016662      -0.693761      -0.469981 -0.692891 -1.103255                  0.604397 -0.105584        1
3    -0.844885 -1.073567      -0.528319      -0.469981  0.123302 -0.494043                 -0.920763 -1.041549        0
4    -1.141852  0.504422      -2.679076       0.831114  0.765836  1.409746                  5.484909 -0.020496        1
```