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»**

Выполнила:
Студентка 3 курса
Группы АС-65
Сунцова М. Д.
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 8

1. Загрузите данные и выведите их статистические характеристики.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler


df = pd.read_csv("pima-indians-diabetes.csv", comment="#", header=None)
df.columns = [
    "Pregnancies", "Glucose", "BloodPressure", "SkinThickness", "Insulin",
    "BMI", "DiabetesPedigreeFunction", "Age", "Outcome"
]


print("Статистические характеристики:")
print(df.describe(), "\n")
```

```
Статистические характеристики:
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin         BMI  DiabetesPedigreeFunction         Age     Outcome
count   768.000000  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   79.799479   31.992578                  0.471876   33.240885    0.348958
std       3.369578   31.972618      19.355807      15.952218  115.244002    7.884160                  0.331329   11.760232    0.476951
min       0.000000    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    0.000000   27.300000                  0.243750   24.000000    0.000000
50%       3.000000  117.000000      72.000000      23.000000   30.500000   32.000000                  0.372500   29.000000    0.000000
75%       6.000000  140.250000      80.000000      32.000000  127.250000   36.600000                  0.626250   41.000000    1.000000
max      17.000000  199.000000     122.000000      99.000000  846.000000   67.100000                  2.420000   81.000000    1.000000
```

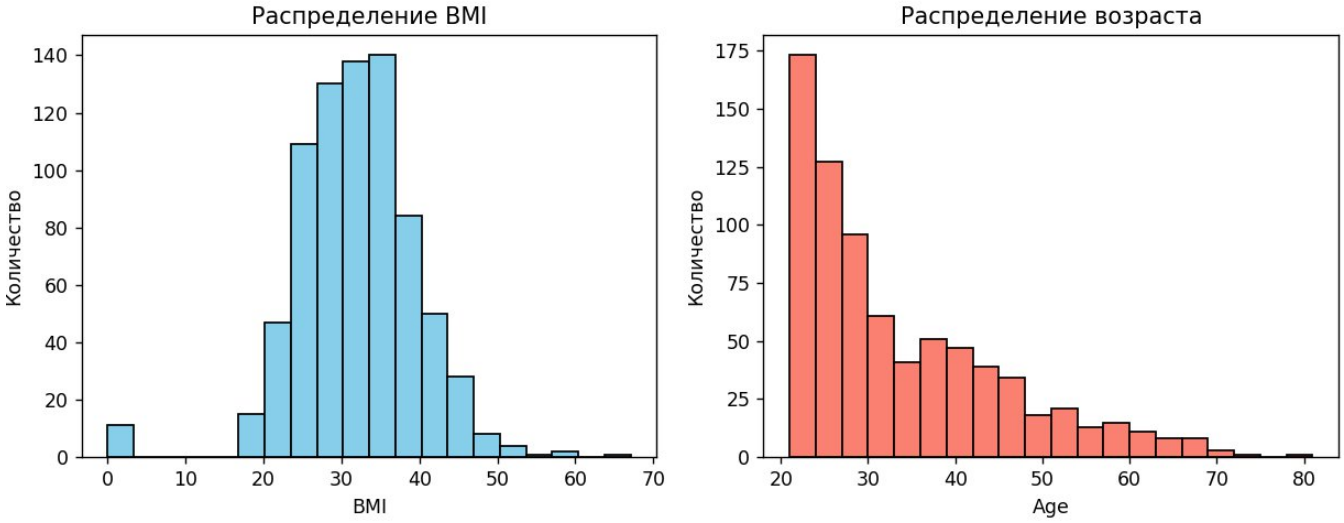
2. Проанализируйте столбцы Glucose, BloodPressure, SkinThickness. Нулевые значения в них, скорее всего, являются ошибками. Замените их медианным значением соответствующего столбца.

```
cols_to_fix = ["Glucose", "BloodPressure", "SkinThickness"]
for col in cols_to_fix:
    median_val = df[col].median()
    df[col] = df[col].replace(0, median_val)


print("Проверка нулевых значений после замены:")
print(df[cols_to_fix].isin([0]).sum(), "\n")
```

```
Проверка нулевых значений после замены:
Glucose          0
BloodPressure    0
SkinThickness    0
dtype: int64
```

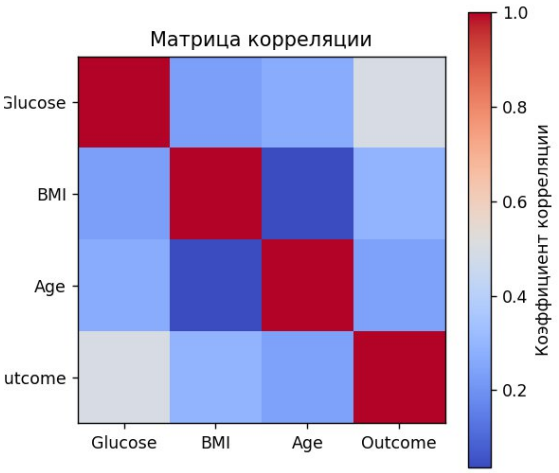
3. Постройте гистограммы для признаков BMI и Age.

```
plt.figure(figsize=(10, 4))
plt.subplot(1, 2, 1)
plt.hist(df["BMI"], bins=20, color="skyblue", edgecolor="black")
plt.title("Распределение BMI")
plt.xlabel("BMI")
plt.ylabel("Количество")


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


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



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

```
corr_matrix = df[["Glucose", "BMI", "Age", "Outcome"]].corr()

plt.figure(figsize=(5, 4))
plt.imshow(corr_matrix, cmap="coolwarm", interpolation="none")
plt.colorbar(label="Коэффициент корреляции")
plt.xticks(range(len(corr_matrix.columns)), corr_matrix.columns)
plt.yticks(range(len(corr_matrix.columns)), corr_matrix.columns)
plt.title("Матрица корреляции")
plt.show()
```



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

```
outcome_counts = df["Outcome"].value_counts()
labels = ["Без диабета", "С диабетом"]
plt.figure(figsize=(5, 5))
plt.pie(outcome_counts, labels=labels, autopct="%.1f%%", startangle=90,
colors=["lightgreen", "lightcoral"])
plt.title("Распределение наличия диабета")
plt.show()
```

6. Примените стандартизацию ко всем признакам, кроме Outcome.

```
# 6
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("Первые строки стандартизированных данных:")
print(df_scaled.head())
```

```
Первые строки стандартизированных данных:
   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
```

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.