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 курса
Группы АС-65
Лопато А.В.
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 1

Выборка Titanic. Содержит информацию о пассажирах лайнера, включая их возраст, пол, класс каюты и факт выживания.

Задачи:

1. Загрузите данные и выведите первые 5 строк, а также общую информацию о столбцах (.info()).
2. Найдите и визуализируйте количество выживших и погибших пассажиров с помощью столбчатой диаграммы.
3. Обработайте пропуски в столбце Age, заполнив их медианным значением.

4. Преобразуйте категориальные признаки Sex и Embarked в числовые с помощью One-Hot Encoding.
5. Постройте гистограмму распределения возрастов пассажиров.
6. Создайте новый признак FamilySize путем сложения значений из столбцов SibSp и Parch.

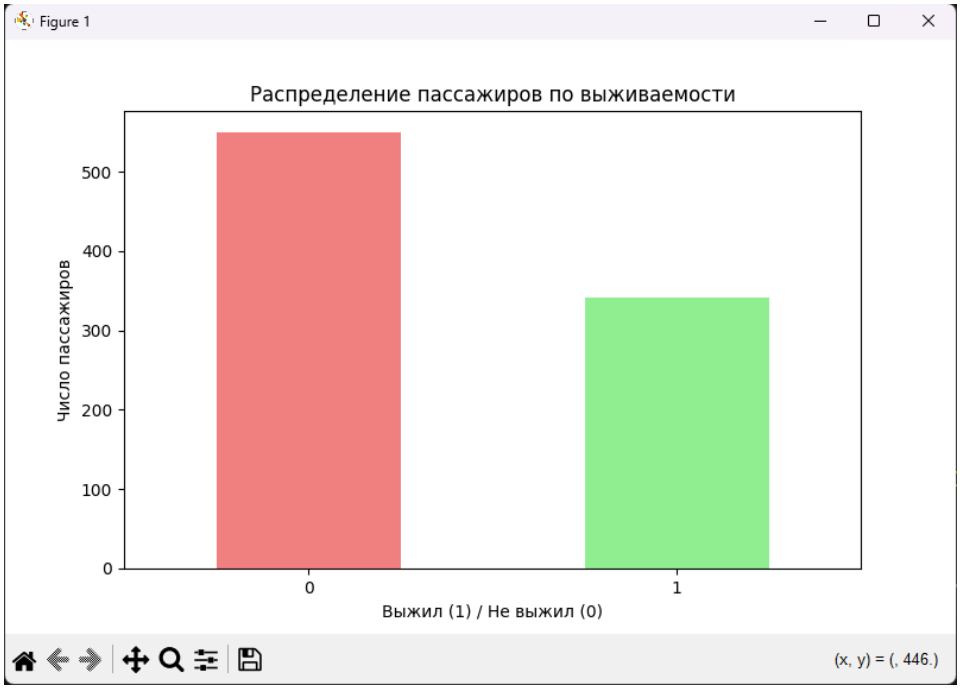
Код программы:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Загрузка данных
df = pd.read_csv("Titanic-Dataset.csv")

# Просмотр данных
print("Первые 5 записей:")
print(df.head())

print("\nСведения о данных:")
print(df.info())

print("\nРаспределение по выживаемости:")
survival_counts = df['Survived'].value_counts()
print(survival_counts)
```

Первые 5 записей:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

```
# Визуализация выживаемости
plt.figure(figsize=(8, 5))
survival_counts.plot(kind='bar', color=['lightcoral', 'lightgreen'])
plt.title("Распределение пассажиров по выживаемости")
plt.xlabel("Выжил (1) / Не выжил (0)")
plt.ylabel("Число пассажиров")
plt.xticks(rotation=0)
plt.show()
```



```
# Обработка пропущенных значений в возрасте
print(f"\nПропущенных значений в возрасте до обработки:
{df['Age'].isna().sum()}")

median_age = df['Age'].median()
df['Age'] = df['Age'].fillna(median_age)

print(f"Пропущенных значений в возрасте после обработки:
{df['Age'].isna().sum()}")
```

```
Распределение по выживаемости:
Survived
0    549
1    342
Name: count, dtype: int64

Пропущенных значений в возрасте до обработки: 177
Пропущенных значений в возрасте после обработки: 0
```

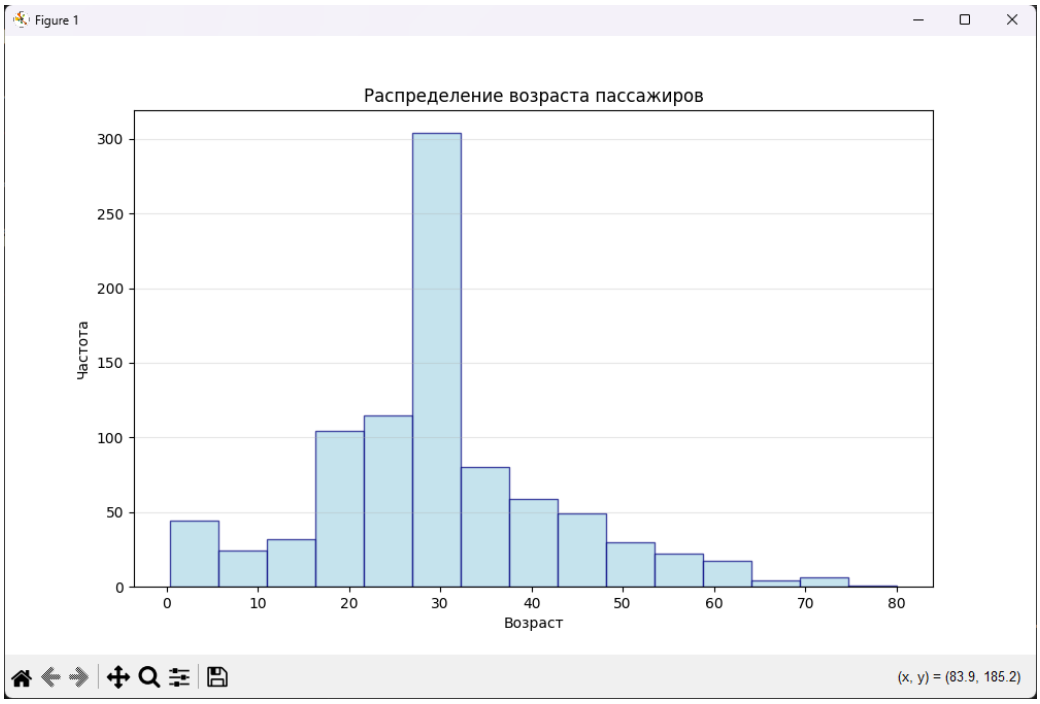
```
Сведения о данных:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
None
```

```
# Преобразование категориальных переменных
categorical_cols = ['Sex', 'Embarked']
df = pd.get_dummies(df, columns=categorical_cols, drop_first=True)

print("\nДанные после преобразования категориальных переменных:")
print(df.head())
```

```
Данные после преобразования категориальных переменных:
   Sex_male  Sex_female  Embarked_C  Embarked_Q  Embarked_S
0      True       False       False       False        True
1     False        True        True       False       False
2     False        True       False       False        True
3     False        True       False       False        True
4      True       False       False       False        True
5      True       False       False        True       False
6      True       False       False       False        True
7      True       False       False       False        True
8     False        True       False       False        True
9     False        True        True       False       False

--- Объяснение преобразованных переменных ---
Sex_male = 1 если мужчина, 0 если женщина
Sex_female = 1 если женщина, 0 если мужчина
Embarked_C = 1 если порт Cherbourg, иначе 0
Embarked_Q = 1 если порт Queenstown, иначе 0
Embarked_S = 1 если порт Southampton, иначе 0
```

```
Пример данных с новым признаком размера семьи:
   SibSp  Parch  FamilySize
0      1      0           2
1      1      0           2
2      0      0           1
3      1      0           2
4      0      0           1
5      0      0           1
6      0      0           1
7      3      1           5
```

```
# Распределение возраста
plt.figure(figsize=(10, 6))
plt.hist(df['Age'], bins=15, color='lightblue', edgecolor='navy',
alpha=0.7)
plt.title("Распределение возраста пассажиров")
plt.xlabel("Возраст")
plt.ylabel("Частота")
plt.grid(axis='y', alpha=0.3)
plt.show()
```




```
# Создание нового признака
df['FamilySize'] = df['SibSp'] + df['Parch'] + 1  # +1 для учета самого пассажира

print("\nПример данных с новым признаком размера семьи:")
print(df[['SibSp', 'Parch', 'FamilySize']].head(8))
```

```
Данные после преобразования категориальных переменных:
   PassengerId  Survived  Pclass                                               Name   Age  SibSp  Parch            Ticket     Fare Cabin  Sex_male  Embarked_Q  Embarked_S
0            1         0       3                            Braund, Mr. Owen Harris  22.0      1      0         A/5 21171   7.2500   NaN      True       False        True
1            2         1       1  Cumings, Mrs. John Bradley (Florence Briggs Th...  38.0      1      0          PC 17599  71.2833   C85     False       False       False
2            3         1       3                             Heikkinen, Miss. Laina  26.0      0      0  STON/O2. 3101282   7.9250   NaN     False       False        True
3            4         1       1       Futrelle, Mrs. Jacques Heath (Lily May Peel)  35.0      1      0            113803  53.1000  C123     False       False        True
4            5         0       3                           Allen, Mr. William Henry  35.0      0      0            373450   8.0500   NaN      True       False        True

Пример данных с новым признаком размера семьи:
   SibSp  Parch  FamilySize
0      1      0           2
1      1      0           2
2      0      0           1
3      1      0           2
4      0      0           1
5      0      0           1
6      0      0           1
7      3      1           5
```

Вывод: в результате выполнения данной лабораторной работы получили практические навыки работы с данными с использованием библиотек Pandas для манипуляции и Matplotlib для визуализации.