

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

Лабораторная работа №1
По дисциплине: «ОМО»
Тема:» Знакомство с анализом данных: предварительная обработка и
визуализация.»

Выполнил:
Студент 3-го курса
Группы АС-66
Савинец М.Д.
Проверил:
Крощенко А.А.

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 11

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.

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

plt.style.use("seaborn-v0_8") # аккуратный стиль графиков

# ===== ЗАДАНИЕ 1 =====
print("\n" + "="*30)
print("ЗАДАНИЕ 1: Загрузка данных и первичный анализ")
print("="*30)

df = pd.read_csv("titanic.csv")
print("\nПервые 5 строк:")
print(df.head().to_string(index=False))
print("\nИнформация о данных:")
print(df.info())

# ===== ЗАДАНИЕ 2 =====
print("\n" + "="*30)
print("ЗАДАНИЕ 2: Визуализация количества выживших и погибших")
print("="*30)

survived_counts = df['Survived'].value_counts()
plt.figure(figsize=(7,5))
bars = plt.bar(['Погибшие', 'Выжившие'], survived_counts, color=['#d9534f',
'#5cb85c'])
plt.title("Распределение выживших и погибших пассажиров", fontsize=14,
fontweight='bold')
plt.ylabel("Количество пассажиров", fontsize=12)
plt.xlabel("Статус", fontsize=12)

for bar in bars:
```

```

        yval = bar.get_height()
        plt.text(bar.get_x() + bar.get_width()/2, yval + 10, str(yval),
                  ha='center', va='bottom', fontsize=11)

plt.tight_layout()
plt.show()

# ===== ЗАДАНИЕ 3 =====
print("\n" + "="*30)
print("ЗАДАНИЕ 3: Обработка пропусков в Age (заполнение медианой)")
print("="*30)

print("Количество пропусков в Age до обработки:", df['Age'].isna().sum())
median_age = df['Age'].median()
df['Age'] = df['Age'].fillna(median_age)
print("Количество пропусков в Age после обработки:", df['Age'].isna().sum())
print(f"Медианное значение, использованное для заполнения: {median_age:.1f}")

# ===== ЗАДАНИЕ 4 =====
# One-Hot Encoding без удаления первой категории
df = pd.get_dummies(df, columns=['Sex', 'Embarked'], drop_first=False)

# Приводим к типу int, чтобы избежать True/False
for col in df.columns:
    if col.startswith('Sex_') or col.startswith('Embarked_'):
        df[col] = df[col].astype(int)

# Переименовываем колонки для читаемости
df.rename(columns={
    'Sex_female': 'Sex_Female',
    'Sex_male': 'Sex_Male',
    'Embarked_C': 'Embarked_C',
    'Embarked_Q': 'Embarked_Q',
    'Embarked_S': 'Embarked_S'
}, inplace=True)

# Выводим результат
print("Категориальные признаки преобразованы.")
print("Новые признаки:", [col for col in df.columns if 'Sex_' in col or
                          'Embarked_' in col])

print("\nПервые строки после кодирования:")
print(df[['Sex_Female', 'Sex_Male', 'Embarked_C', 'Embarked_Q',
          'Embarked_S']].head(10).to_string(index=False))

# ===== ЗАДАНИЕ 5 =====
print("\n" + "="*30)
print("ЗАДАНИЕ 5: Гистограмма распределения возрастов")
print("="*30)

plt.figure(figsize=(7,5))
plt.hist(df['Age'], bins=20, color='#5bc0de', edgecolor='black')
plt.title("Распределение возрастов пассажиров", fontsize=14,
          fontweight='bold')
plt.xlabel("Возраст", fontsize=12)
plt.ylabel("Количество пассажиров", fontsize=12)

```

```
plt.grid(alpha=0.3)
plt.tight_layout()
plt.show()

# ===== ЗАДАНИЕ 6 =====
print("\n" + "="*30)
print("ЗАДАНИЕ 6: Новый признак FamilySize")
print("="*30)

df['FamilySize'] = df['SibSp'] + df['Parch']
print("Пример новых данных:")
print(df[['SibSp', 'Parch', 'FamilySize']].head().to_string(index=False))

# ===== СОХРАНЕНИЕ =====
print("\n" + "="*30)
print("СОХРАНЕНИЕ ОБРАБОТАННОГО DATASETА")
print("="*30)

df.to_csv("titanic_after.csv", index=False)
print("Файл titanic_after.csv сохранён с обработанными данными.")
```

Задание 1.

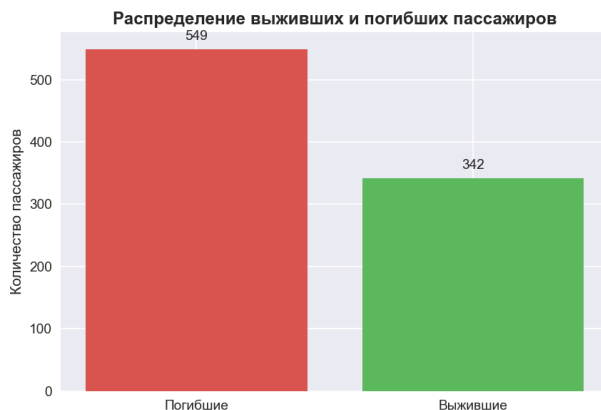
Первые 5 строк:

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

Информация 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
```

Задание 2.



Задание 3.

Количество пропусков в Age до обработки: 177

Количество пропусков в Age после обработки: 0

Медианное значение, использованное для заполнения: 28.0

Задание 4.

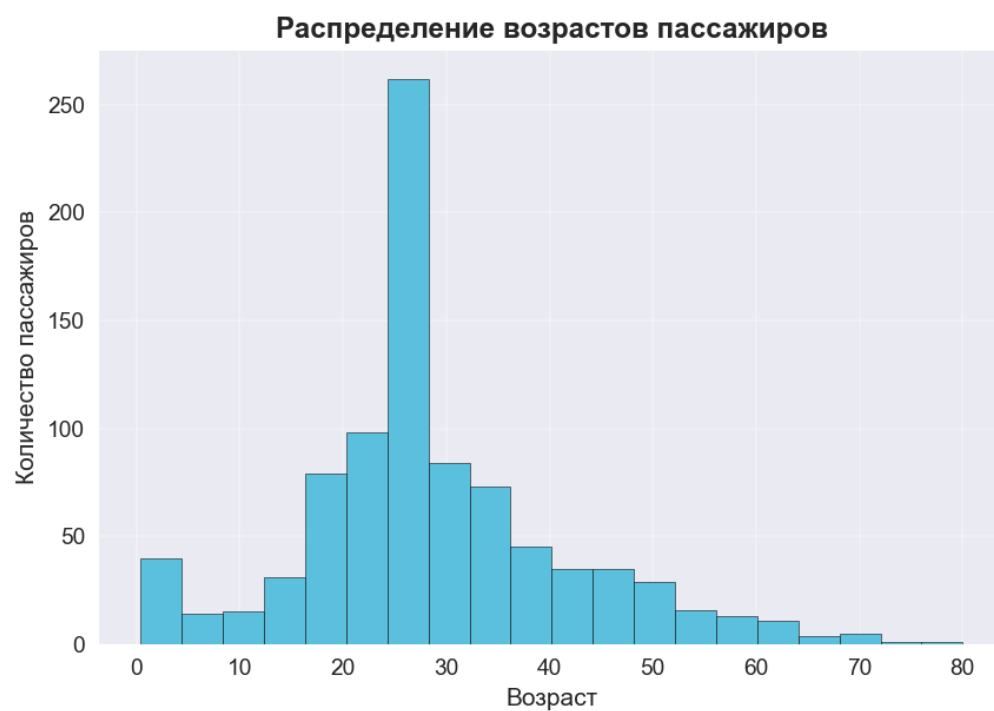
Категориальные признаки преобразованы.

Новые признаки: ['Sex_Female', 'Sex_Male', 'Embarked_C', 'Embarked_Q', 'Embarked_S']

Первые строки после кодирования:

Sex_Female	Sex_Male	Embarked_C	Embarked_Q	Embarked_S
0	1	0	0	1
1	0	1	0	0
1	0	0	0	1
1	0	0	0	1
0	1	0	0	1
0	1	0	1	0
0	1	0	0	1
0	1	0	0	1
1	0	0	0	1
1	0	1	0	0

Задание 5.



Задание 6.

Пример новых данных:

SibSp	Parch	FamilySize
1	0	1
1	0	1
0	0	0
1	0	1
0	0	0

Вывод: в результате выполнения данной лабораторной работы 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.