

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
Шлейхер А. С.
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 10

Выборка German Credit Data. Содержит информацию о заемщиках, включая их кредитную историю, цель кредита, возраст, и оценку кредитоспособности (хороший/плохой).

Задачи:

1. Загрузите данные и выведите информацию о них.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler

df = pd.read_csv("german_credit.csv")
print("Информация о данных:")
print(df.info(), "\n")
```

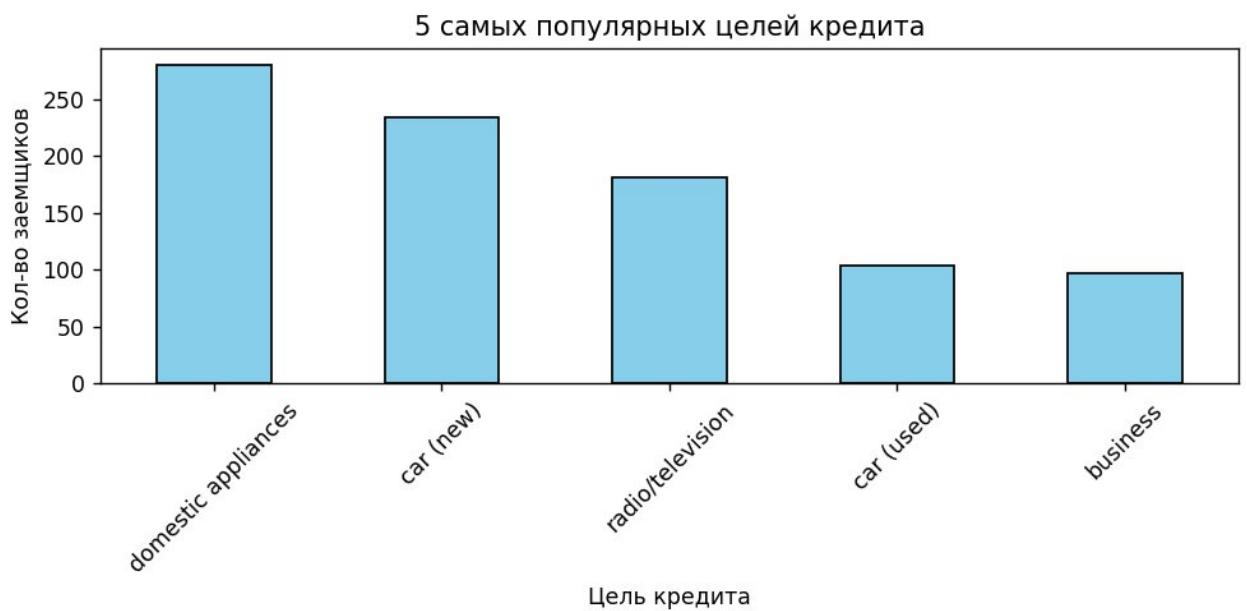
```
Информация о данных:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   default          1000 non-null    int64  
 1   account_check_status  1000 non-null    object  
 2   duration_in_month    1000 non-null    int64  
 3   credit_history       1000 non-null    object  
 4   purpose             1000 non-null    object  
 5   credit_amount        1000 non-null    int64  
 6   savings             1000 non-null    object  
 7   present_emp_since   1000 non-null    object  
 8   installment_as_income_perc  1000 non-null    int64  
 9   personal_status_sex  1000 non-null    object  
 10  other_debtors       1000 non-null    object  
 11  present_res_since   1000 non-null    int64  
 12  property            1000 non-null    object  
 13  age                 1000 non-null    int64  
 14  other_installment_plans  1000 non-null    object  
 15  housing             1000 non-null    object  
 16  credits_this_bank   1000 non-null    int64  
 17  job                 1000 non-null    object  
 18  people_under_maintenance  1000 non-null    int64  
 19  telephone           1000 non-null    object  
 20  foreign_worker      1000 non-null    object  
dtypes: int64(8), object(13)
memory usage: 164.2+ KB
None
```

2. Проанализируйте распределение цели кредита (Purpose). Визуализируйте 5 самых популярных целей.

```
print("Цели кредита:")
print(df["purpose"].value_counts(), "\n")

plt.figure(figsize=(8, 4))
df["purpose"].value_counts().head(5).plot(kind="bar", color="skyblue",
edgecolor="black")
plt.title("5 самых популярных целей кредита")
plt.xlabel("Цель кредита")
plt.ylabel("Кол-во заемщиков")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
Цели кредита:
purpose
domestic appliances    280
car (new)               234
radio/television        181
car (used)              103
business                97
(vacation - does not exist?) 50
education               22
repairs                 12
furniture/equipment     12
retraining               9
Name: count, dtype: int64
```



3. Преобразуйте категориальные признаки Sex и Housing в числовой формат .

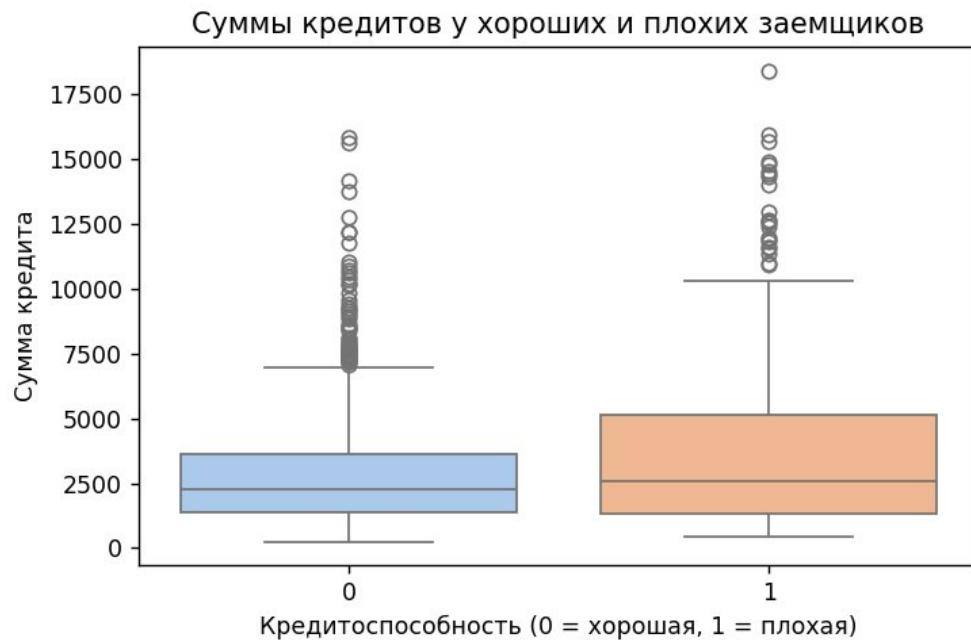
```
df["sex"] = df["personal_status_sex"].apply(lambda x: 0 if "female" in x.lower() else 1)
print("Уникальные значения sex:", df["sex"].unique(),)

df["housing"] = df["housing"].map({"own": 2, "rent": 1, "for free": 0})
print("Уникальные значения housing:", df["housing"].unique(), "\n")
```

```
Уникальные значения sex: [1 0]
Уникальные значения housing: [2 0 1]
```

4. Постройте "ящик с усами" для Credit amount, чтобы сравнить суммы кредитов у "хороших" и "плохих" заемщиков.

```
plt.figure(figsize=(6, 4))
sns.boxplot(data=df, x="default", y="credit_amount", hue="default",
palette="pastel", legend=False)
plt.title("Суммы кредитов у хороших и плохих заемщиков")
plt.xlabel("Кредитоспособность (0 = хорошая, 1 = плохая)")
plt.ylabel("Сумма кредита")
plt.tight_layout()
plt.show()
```



5. Создайте сводную таблицу, показывающую средний возраст (Age) и среднюю длительность кредита (Duration) для каждой категории кредитной истории (Credit history).

```
pivot = df.pivot_table(
    values=["age", "duration_in_month"],
    index="credit_history",
    aggfunc="mean"
)
```

```
print("Средний возраст и длительность кредита по кредитной истории:")
print(pivot, "\n")
```

| Средний возраст и длительность кредита по кредитной истории: | age | duration_in_month |
|--|-----------|-------------------|
| credit_history | | |
| all credits at this bank paid back duly | 36.265306 | 22.693878 |
| critical account/ other credits existing (not a...) | 38.436860 | 19.488055 |
| delay in paying off in the past | 36.136364 | 26.215909 |
| existing credits paid back duly till now | 33.877358 | 20.111321 |
| no credits taken/ all credits paid back duly | 34.300000 | 27.875000 |

6. Нормализуйте числовые столбцы Age, Credit amount, Duration.

```
scaler = MinMaxScaler()
num_cols = ["age", "credit_amount", "duration_in_month"]
df_norm = df.copy()
df_norm[num_cols] = scaler.fit_transform(df[num_cols])

print("Первые строки после нормализации числовых признаков:")
print(df_norm[num_cols].head())
```

| Первые строки после нормализации числовых признаков: | | |
|--|----------|---------------|
| | age | credit_amount |
| 0 | 0.857143 | 0.050567 |
| 1 | 0.053571 | 0.313690 |
| 2 | 0.535714 | 0.101574 |
| 3 | 0.464286 | 0.419941 |
| 4 | 0.607143 | 0.254209 |

Вывод: получила практические навыки работы с данными с использованием библиотек Pandas для манипуляции и Matplotlib для визуализации. Научилась выполнять основные шаги предварительной обработки данных, такие как очистка, нормализация и работа с различными типами признаков.