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
Макарский А.Э.
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

Задание 1. Загрузите данные и выведите информацию о них.

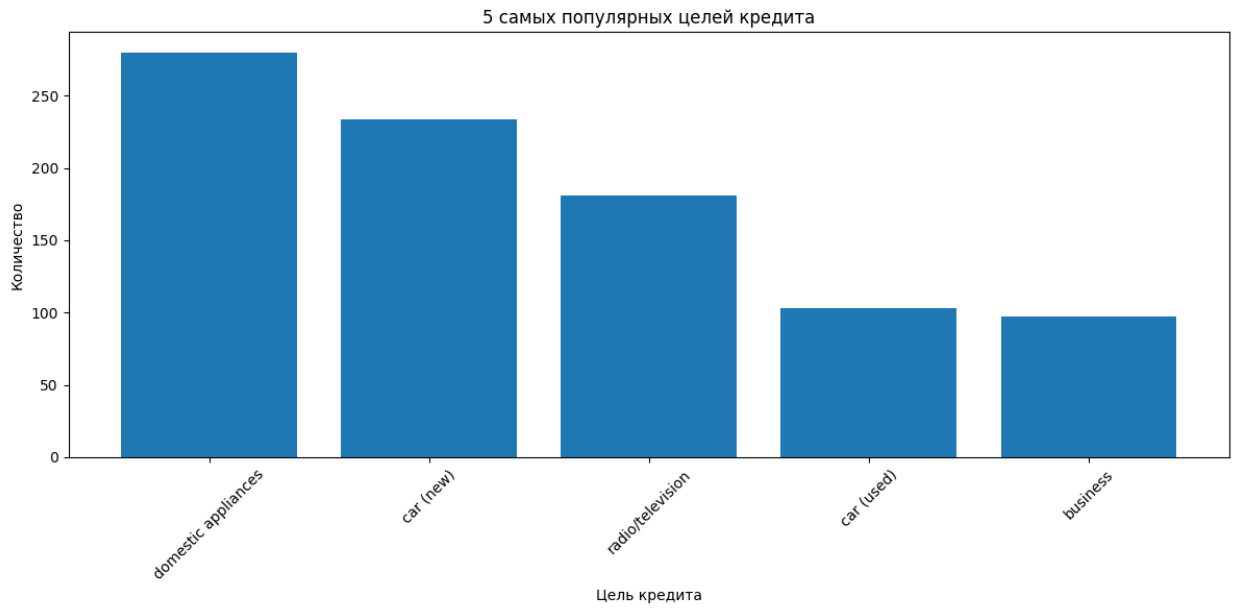
```
df = pd.read_csv('german_credit.csv')
print("=== ИНФОРМАЦИЯ О ДАННЫХ ===")
print(f"Размер датасета: {df.shape}")
print("\nПервые 5 строк:")
print(df.head())
print("\nИнформация о типах данных:")
print(df.info())
print("\nСтатистическое описание числовых признаков:")
print(df.describe())
print("\nПроверка пропущенных значений:")
print(df.isnull().sum())
```

```
=== ИНФОРМАЦИЯ О ДАННЫХ ===
Размер датасета: (1000, 21)

Первые 5 строк:
   default account_check_status  duration_in_month  \
0        0               < 0 DM                  6
1        1    0 <= ... < 200 DM                 48
2        0  no checking account                 12
3        0               < 0 DM                 42
4        1               < 0 DM                 24

                                  credit_history  \
0  critical account/ other credits existing (not ...
1          existing credits paid back duly till now
2  critical account/ other credits existing (not ...
3          existing credits paid back duly till now
4                  delay in paying off in the past
```

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

```
print("\n=== РАСПРЕДЕЛЕНИЕ ЦЕЛИ КРЕДИТА ===")
purpose_counts = df['purpose'].value_counts()
print("Распределение целей кредита:")
print(purpose_counts)

plt.figure(figsize=(12, 6))
top_5_purposes = purpose_counts.head(5)
plt.bar(top_5_purposes.index, top_5_purposes.values)
plt.title('5 самых популярных целей кредита')
plt.xlabel('Цель кредита')
plt.ylabel('Количество')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
=== РАСПРЕДЕЛЕНИЕ ЦЕЛИ КРЕДИТА ===
Распределение целей 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
```



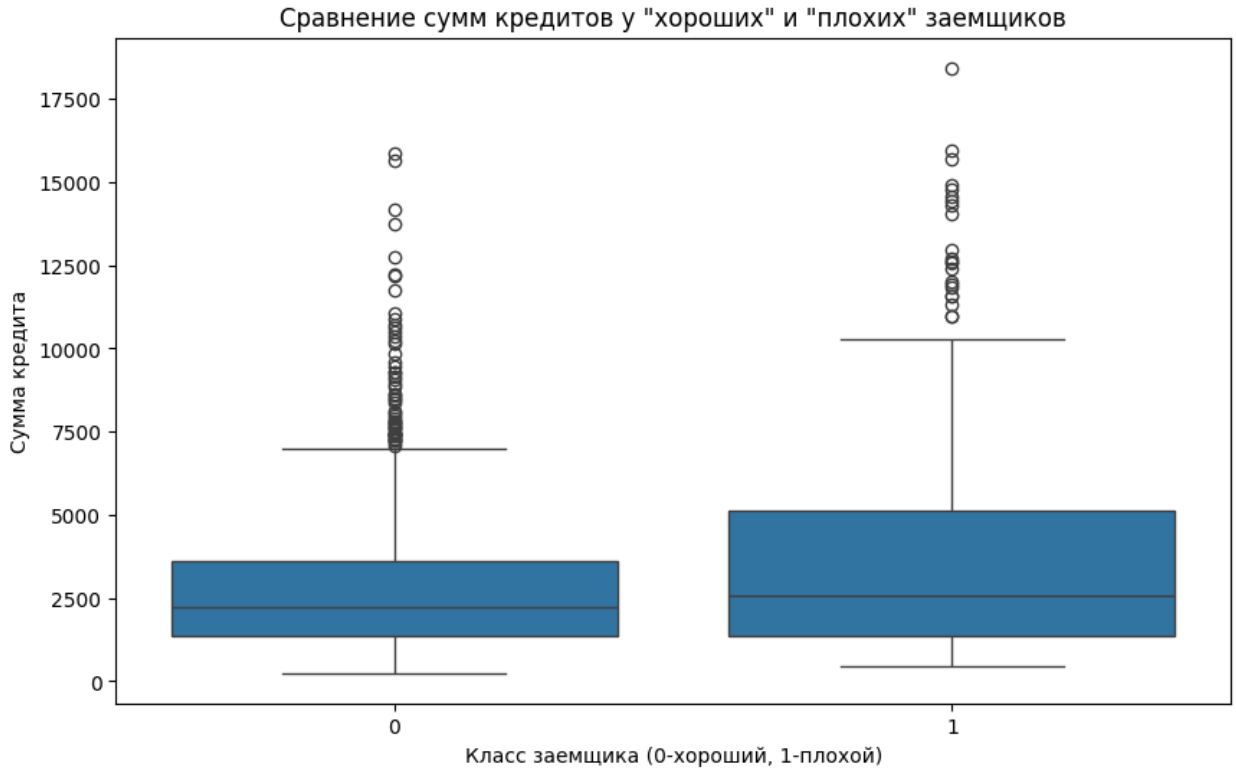
Задание 3. Преобразуйте категориальные признаки Sex и Housing в числовой формат.

```
print("\n=== ПРЕОБРАЗОВАНИЕ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ ===")

# Преобразование Sex (из personal_status_sex)
print("Уникальные значения personal_status_sex:")
print(df['personal_status_sex'].unique())

# Создаем бинарный признак для пола
df['sex_encoded'] = df['personal_status_sex'].apply(
    lambda x: 1 if 'male' in x.lower() else 0
)
print(f"\nРаспределение по полу:")
print(df['sex_encoded'].value_counts())
print("\nУникальные значения housing:")
print(df['housing'].unique())
housing_encoded = pd.get_dummies(df['housing'], prefix='housing')
df = pd.concat([df, housing_encoded], axis=1)
print("\nРезультат One-Hot Encoding для housing:")
print(housing_encoded.head())
```

```
=== ПРЕОБРАЗОВАНИЕ КАТЕГОРИАЛЬНЫХ ПРИЗНАКОВ ===
Уникальные значения personal_status_sex:
['male : single' 'female : divorced/separated/married'
 'male : divorced/separated' 'male : married/widowed']
```

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

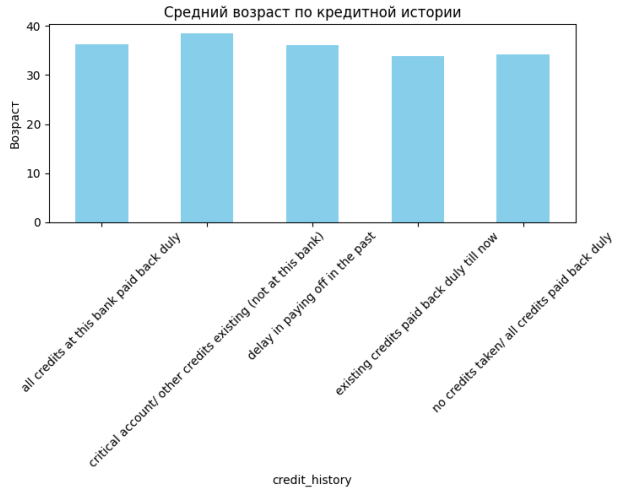
```
print("\n=== СРАВНЕНИЕ СУММ КРЕДИТОВ ===")
plt.figure(figsize=(10, 6))
sns.boxplot(x='default', y='credit_amount', data=df)
plt.title('Сравнение сумм кредитов у "хороших" и "плохих" заемщиков')
plt.xlabel('Класс заемщика (0-хороший, 1-плохой)')
plt.ylabel('Сумма кредита')
plt.show()

good_borrowers = df[df['default'] == 0]['credit_amount']
bad_borrowers = df[df['default'] == 1]['credit_amount']

print(f"Средняя сумма кредита для хороших заемщиков: {good_borrowers.mean():.2f}")
print(f"Средняя сумма кредита для плохих заемщиков: {bad_borrowers.mean():.2f}")
print(f"Медианная сумма кредита для хороших заемщиков: {good_borrowers.median():.2f}")
print(f"Медианная сумма кредита для плохих заемщиков: {bad_borrowers.median():.2f}")
```



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

```
print("\n=== СВОДНАЯ ТАБЛИЦА ===")
pivot_table = df.pivot_table(
    values=['age', 'duration_in_month'],
    index='credit_history',
    aggfunc={'age': 'mean', 'duration_in_month': 'mean'}
).round(2)
print("Средний возраст и длительность кредита по кредитной истории:")
print(pivot_table)

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15, 6))

pivot_table['age'].plot(kind='bar', ax=ax1, color='skyblue')
ax1.set_title('Средний возраст по кредитной истории')
ax1.set_ylabel('Возраст')
ax1.tick_params(axis='x', rotation=45)

pivot_table['duration_in_month'].plot(kind='bar', ax=ax2, color='lightcoral')
ax2.set_title('Средняя длительность кредита по кредитной истории')
ax2.set_ylabel('Длительность (месяцы)')
ax2.tick_params(axis='x', rotation=45)

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

```
=== СВОДНАЯ ТАБЛИЦА ===
Средний возраст и длительность кредита по кредитной истории:
                                                  age  duration_in_month
credit_history
all credits at this bank paid back duly         36.27              22.69
critical account/ other credits existing (not a...  38.44              19.49
delay in paying off in the past                 36.14              26.22
existing credits paid back duly till now        33.88              20.11
no credits taken/ all credits paid back duly    34.30              27.88
```

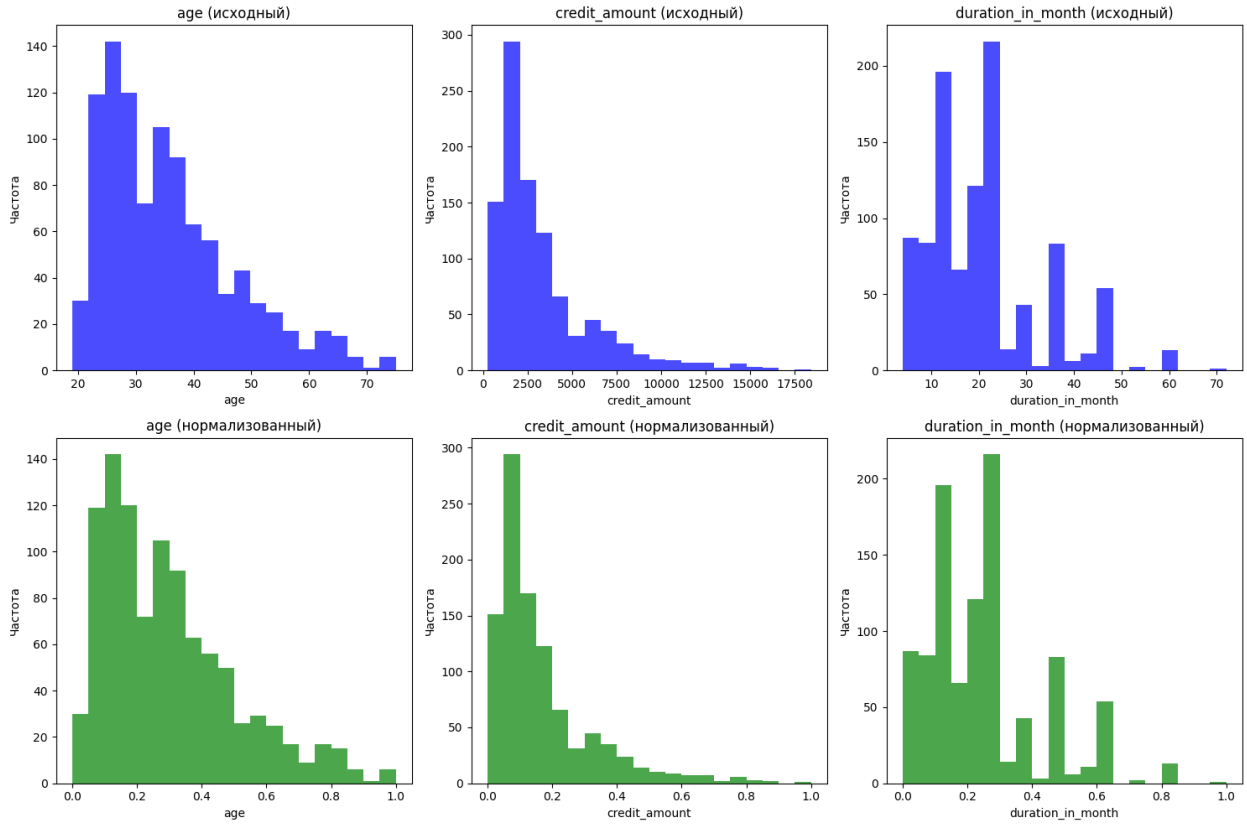


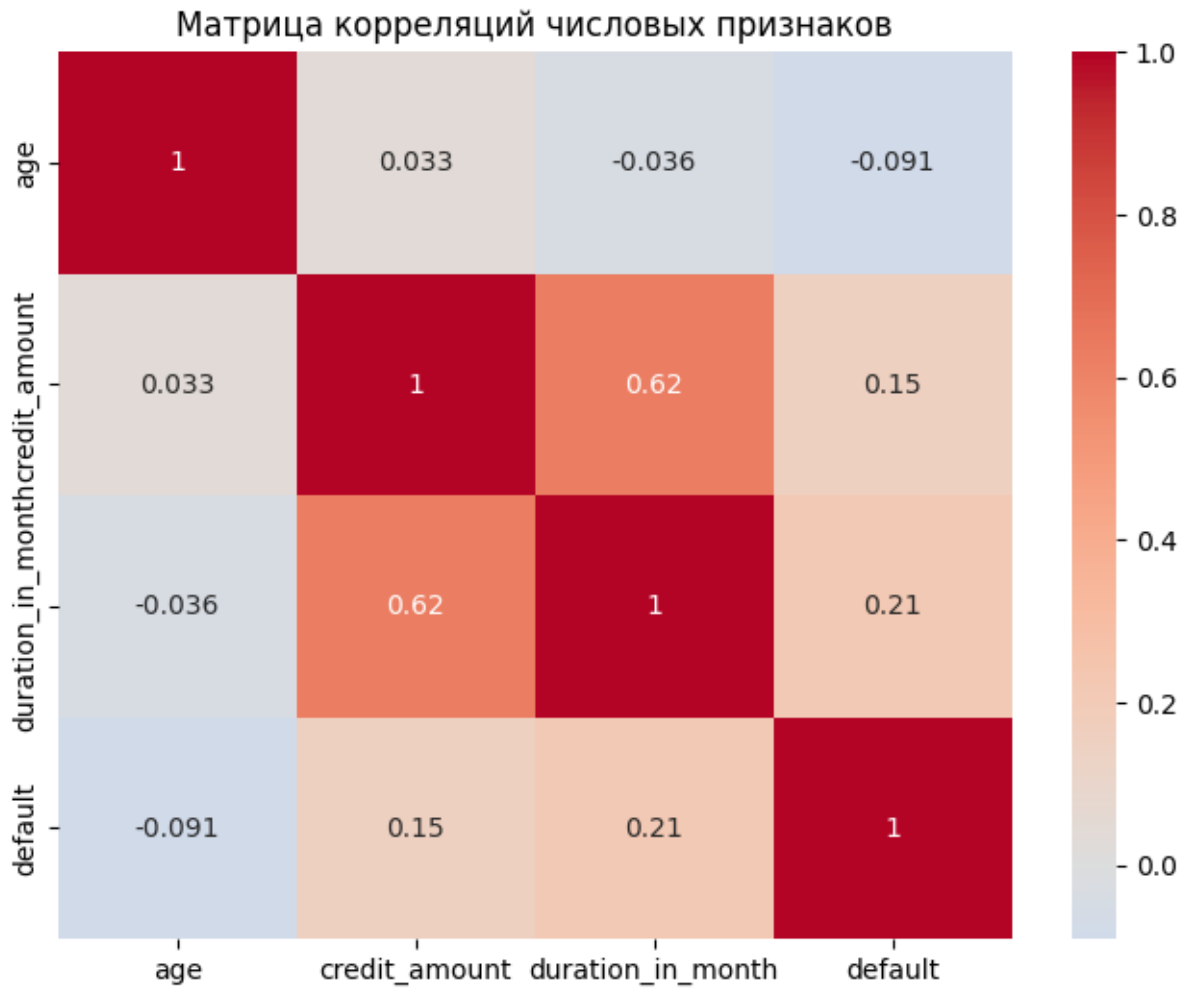
Задание 6. Нормализуйте числовые столбцы Age, Credit amount, Duration.

```
print("\n=== НОРМАЛИЗАЦИЯ ЧИСЛОВЫХ ПРИЗНАКОВ ===")
numeric_columns = ['age', 'credit_amount', 'duration_in_month']

print("Исходные статистики:")
print(df[numeric_columns].describe())
scaler_standard = StandardScaler()
df_standardized = df.copy()
df_standardized[numeric_columns] = scaler_standard.fit_transform(df[numeric_columns])

print("\nПосле стандартизации (Z-score):")
print(df_standardized[numeric_columns].describe())

scaler_minmax = MinMaxScaler()
df_normalized = df.copy()
df_normalized[numeric_columns] = scaler_minmax.fit_transform(df[numeric_columns])

print("\nПосле нормализации (Min-Max):")
print(df_normalized[numeric_columns].describe())

fig, axes = plt.subplots(2, 3, figsize=(15, 10))
for i, col in enumerate(numeric_columns):

    axes[0, i].hist(df[col], bins=20, alpha=0.7, color='blue')
    axes[0, i].set_title(f'{col} (исходный)')
    axes[0, i].set_xlabel(col)
    axes[0, i].set_ylabel('Частота')

    axes[1, i].hist(df_normalized[col], bins=20, alpha=0.7, color='green')
    axes[1, i].set_title(f'{col} (нормализованный)')
    axes[1, i].set_xlabel(col)
    axes[1, i].set_ylabel('Частота')

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

После стандартизации (Z-score):

	age	credit_amount	duration_in_month
count	1.000000e+03	1.000000e+03	1.000000e+03
mean	5.329071e-17	6.661338e-17	1.136868e-16
std	1.000500e+00	1.000500e+00	1.000500e+00
min	-1.455261e+00	-1.070865e+00	-1.402415e+00
25%	-7.516417e-01	-6.754833e-01	-7.386675e-01
50%	-2.239269e-01	-3.373443e-01	-2.408572e-01
75%	5.676451e-01	2.484620e-01	2.569531e-01
max	3.470076e+00	5.370789e+00	4.239436e+00





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.