Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

Отчёт к лабораторным работам по курсу

«Методы машинного обучения»

Лабораторная работа №2 «Обработка признаков (часть 1)»

Выполнил:

студент(ка) группы ИУ5И-21М Лю Бэйбэй

подпись, дата

Проверил:

к.т.н., доц., Виноградовой М.В.

подпись, дата

Москва, 2022 г.

1. описание задания

1. Выбрать набор данных (датасет), содержащий категориальные и числовые признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.) Просьба не использовать датасет, на котором данная задача решалась в лекции.
2. Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи:
    i. устранение пропусков в данных;
    ii. кодирование категориальных признаков;
    iii. нормализация числовых признаков.

2. Текст программы и экранные формы с примерами выполнения программы.

Импортирование необходимых библиотек.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.gridspec as gridspec
import scipy.stats as stats
from sklearn.preprocessing import OneHotEncoder
```

Импортирование данных.

```
df = pd.read_csv("E:\liu\lab\lab2\Customers.csv")
```

```
df.head()
```

	CustomerID	Gender	Age	Annual Income ($)	Spending Score (1-100)	Profession	Work Experience	Family Size
0	1	Male	19	15000	39	Healthcare	1	4
1	2	Male	21	35000	81	Engineer	3	3
2	3	Female	20	86000	6	Engineer	1	1
3	4	Female	23	59000	77	Lawyer	0	2
4	5	Female	31	38000	40	Entertainment	2	6

```
df.shape
```

```
(2000, 8)
```

```
df.isnull().sum()
```

```
CustomerID                0
Gender                    0
Age                       0
Annual Income ($)         0
Spending Score (1-100)    0
Profession               35
Work Experience           0
Family Size               0
dtype: int64
```

Пропуски в данных и устранение пропусков в данных.

```
df.loc[df['Profession'].isnull()]
```

	CustomerID	Gender	Age	Annual Income ($)	Spending Score (1-100)	Profession	Work Experience	Family Size
79	80	Female	49	98000	42	NaN	1	1
118	119	Female	51	84000	43	NaN	2	7
219	220	Female	59	76000	61	NaN	9	1
237	238	Male	95	36000	35	NaN	0	4
437	438	Male	76	136259	14	NaN	0	7

```
df.dropna(axis=0, how='any',inplace=True)
```

Данные после обработки.

```
df.shape
```

```
(1965, 8)
```

```
df.isnull().sum()
```

```
CustomerID               0
Gender                   0
Age                      0
Annual Income ($)        0
Spending Score (1-100)   0
Profession               0
Work Experience          0
Family Size              0
dtype: int64
```

Кодирование категорий наборами бинарных значений - one-hot encoding.

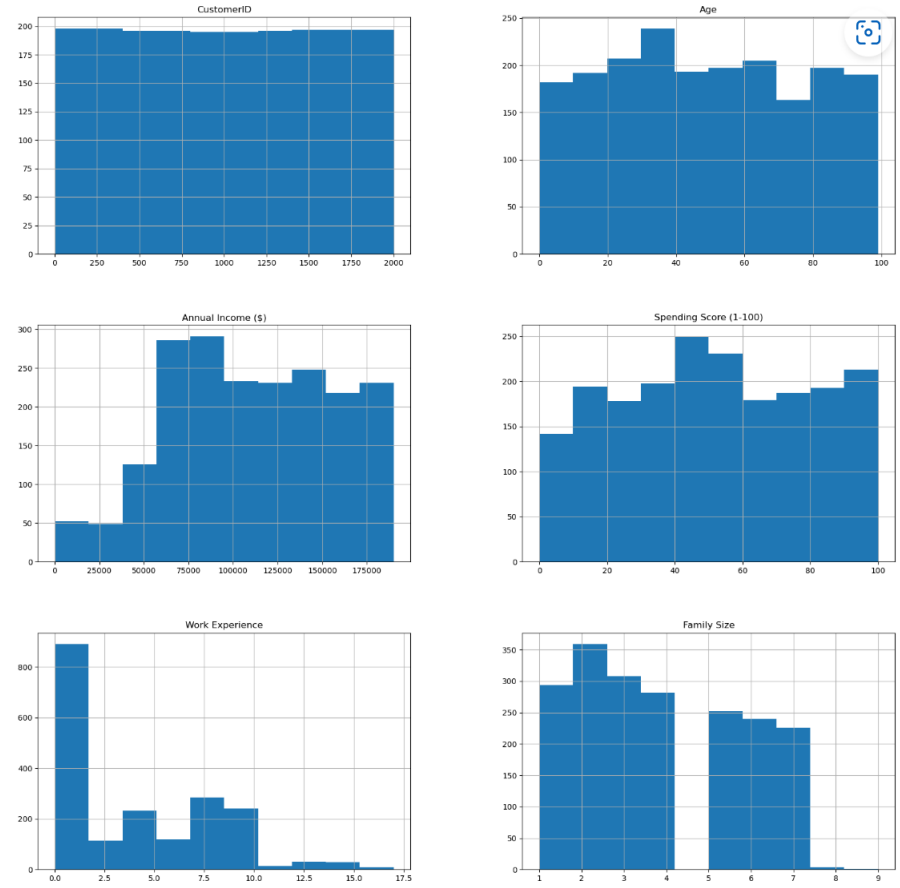
```
gender_ohe.todense()
```

```
matrix([[0., 1.],
        [0., 1.],
        [1., 0.],
        ...,
        [0., 1.],
        [0., 1.],
        [0., 1.]])
```

```
pro_ohe = ohe.fit_transform(df[['Profession']])
pro_ohe.todense()
```

```
matrix([[0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 1., ..., 0., 0., 0.],
        [0., 0., 1., ..., 0., 0., 0.],
        ...,
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.],
        [0., 0., 0., ..., 0., 0., 0.]])
```

Исходное распределение.

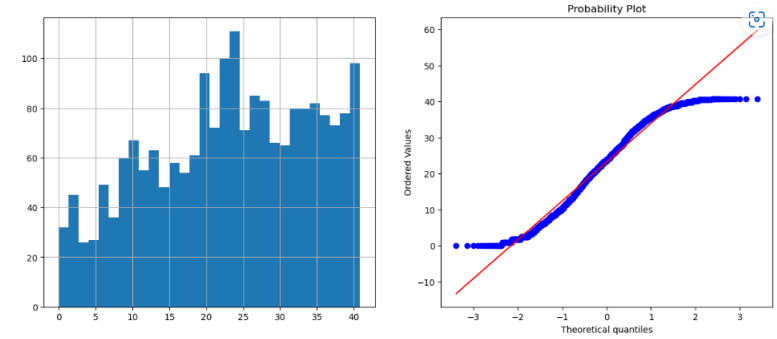


Хороший результат.

Преобразование Бокса-Кокса

```
df['Spending Score (1-100)_boxcox'], param = stats.boxcox(df['Spending Score (1-100)'])
print('Оптимальное значение λ = {}'.format(param))
diagnostic_plots(df, 'Spending Score (1-100)_boxcox')
```

```
Оптимальное значение λ = 0.7498555325041829
```



Преобразование Йео-Джонсона

```
df['Spending Score (1-100)'] = df['Spending Score (1-100)'].astype('float')
df['Spending Score (1-100)_yeojohnson'], param = stats.yeojohnson(df['Spending Score (1-100)'])
print('Оптимальное значение λ = {}'.format(param))
diagnostic_plots(df, 'Spending Score (1-100)_yeojohnson')
```

```
Оптимальное значение λ = 0.7565367536857261
```