Рубежный контроль №1, Черников Анатолий РТ5-61Б

Задача 3, Вариант 4 Для заданного набора данных произведите масштабирование данных (для одного признака) и преобразование категориальных признаков в количественные двумя способами (label encoding, one hot encoding) для одного признака. Какие методы Вы использовали для решения задачи и почему? Дополнительные требования по группам: Для студентов группы РТ5-61Б - для пары произвольных колонок данных построить график "Jointplot".

Ввод [4]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, MinMaxScaler
```

Загрузка датасета и вывод общей информации о нём

Ввод [5]:

```
data = pd.read_csv("states_all.csv")
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1715 entries, 0 to 1714
Data columns (total 25 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   PRIMARY_KEY                     1715 non-null   object
 1   STATE                           1715 non-null   object
 2   YEAR                            1715 non-null   int64
 3   ENROLL                          1224 non-null   float64
 4   TOTAL_REVENUE                   1275 non-null   float64
 5   FEDERAL_REVENUE                 1275 non-null   float64
 6   STATE_REVENUE                   1275 non-null   float64
 7   LOCAL_REVENUE                   1275 non-null   float64
 8   TOTAL_EXPENDITURE               1275 non-null   float64
 9   INSTRUCTION_EXPENDITURE         1275 non-null   float64
 10  SUPPORT_SERVICES_EXPENDITURE    1275 non-null   float64
 11  OTHER_EXPENDITURE               1224 non-null   float64
 12  CAPITAL_OUTLAY_EXPENDITURE      1275 non-null   float64
 13  GRADES_PK_G                     1542 non-null   float64
 14  GRADES_KG_G                     1632 non-null   float64
 15  GRADES_4_G                      1632 non-null   float64
 16  GRADES_8_G                      1632 non-null   float64
 17  GRADES_12_G                     1632 non-null   float64
 18  GRADES_1_8_G                    1020 non-null   float64
 19  GRADES_9_12_G                   1071 non-null   float64
 20  GRADES_ALL_G                    1632 non-null   float64
 21  AVG_MATH_4_SCORE                565 non-null    float64
 22  AVG_MATH_8_SCORE                602 non-null    float64
 23  AVG_READING_4_SCORE             650 non-null    float64
 24  AVG_READING_8_SCORE             562 non-null    float64
dtypes: float64(22), int64(1), object(2)
memory usage: 335.1+ KB
```

Ввод [6]:

```
data.head()
```

Out[6]:

	PRIMARY_KEY	STATE	YEAR	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	ST
0	1992_ALABAMA	ALABAMA	1992	NaN	2678885.0	304177.0	
1	1992_ALASKA	ALASKA	1992	NaN	1049591.0	106780.0	
2	1992_ARIZONA	ARIZONA	1992	NaN	3258079.0	297888.0	
3	1992_ARKANSAS	ARKANSAS	1992	NaN	1711959.0	178571.0	
4	1992_CALIFORNIA	CALIFORNIA	1992	NaN	26260025.0	2072470.0	

5 rows × 25 columns

Ввод [7]:

```
data.dtypes
```

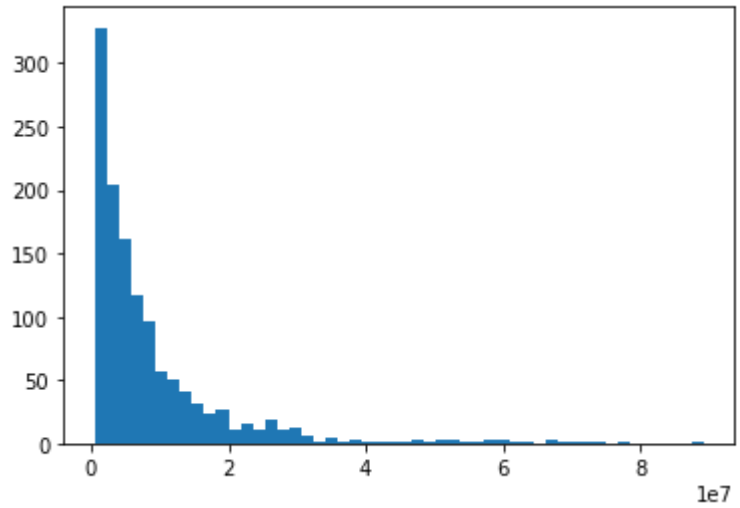
Out[7]:

```
PRIMARY_KEY                    object
STATE                          object
YEAR                            int64
ENROLL                        float64
TOTAL_REVENUE                 float64
FEDERAL_REVENUE               float64
STATE_REVENUE                 float64
LOCAL_REVENUE                 float64
TOTAL_EXPENDITURE             float64
INSTRUCTION_EXPENDITURE       float64
SUPPORT_SERVICES_EXPENDITURE  float64
OTHER_EXPENDITURE             float64
CAPITAL_OUTLAY_EXPENDITURE    float64
GRADES_PK_G                   float64
GRADES_KG_G                   float64
GRADES_4_G                    float64
GRADES_8_G                    float64
GRADES_12_G                   float64
GRADES_1_8_G                  float64
GRADES_9_12_G                 float64
GRADES_ALL_G                  float64
AVG_MATH_4_SCORE              float64
AVG_MATH_8_SCORE              float64
AVG_READING_4_SCORE           float64
AVG_READING_8_SCORE           float64
dtype: object
```

MinMax масштабирование признака TOTAL_REVENUE

Ввод [8]:

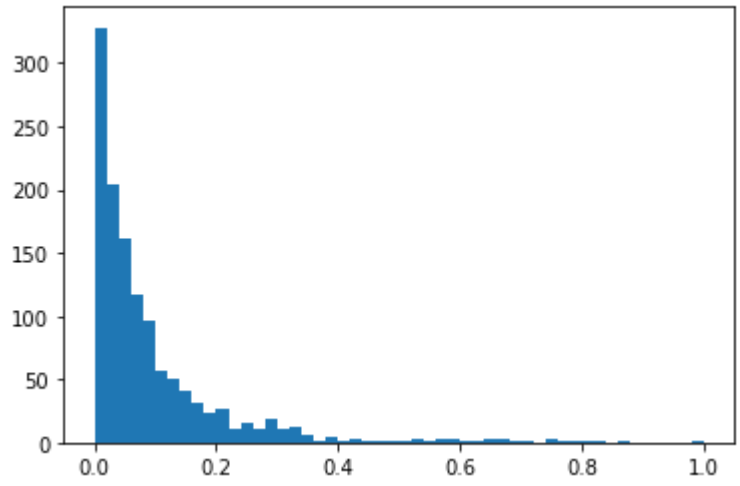
```
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data[['TOTAL_REVENUE']])
plt.hist(data['TOTAL_REVENUE'],50)
plt.show()
```



Как видно, график не изменился, а значения теперь лежат в промежутке от 0 до 1

Ввод [9]:

```
plt.hist(sc1_data, 50)
plt.show()
```



Проведём преобразование категориального признака STATE в количественный посредством label encoding

Ввод [10]:

```
state_data = data[['STATE']]
state_data.head()
```

Out[10]:

	STATE
0	ALABAMA
1	ALASKA
2	ARIZONA
3	ARKANSAS
4	CALIFORNIA

Ввод [11]:

```
state_data['STATE'].unique()
```

Out[11]:

```
array(['ALABAMA', 'ALASKA', 'ARIZONA', 'ARKANSAS', 'CALIFORNIA',
       'COLORADO', 'CONNECTICUT', 'DELAWARE', 'DISTRICT_OF_COLUMBIA',
       'FLORIDA', 'GEORGIA', 'HAWAII', 'IDAHO', 'ILLINOIS', 'INDIANA',
       'IOWA', 'KANSAS', 'KENTUCKY', 'LOUISIANA', 'MAINE', 'MARYLAND',
       'MASSACHUSETTS', 'MICHIGAN', 'MINNESOTA', 'MISSISSIPPI',
       'MISSOURI', 'MONTANA', 'NEBRASKA', 'NEVADA', 'NEW_HAMPSHIRE',
       'NEW_JERSEY', 'NEW_MEXICO', 'NEW_YORK', 'NORTH_CAROLINA',
       'NORTH_DAKOTA', 'OHIO', 'OKLAHOMA', 'OREGON', 'PENNSYLVANIA',
       'RHODE_ISLAND', 'SOUTH_CAROLINA', 'SOUTH_DAKOTA', 'TENNESSEE',
       'TEXAS', 'UTAH', 'VERMONT', 'VIRGINIA', 'WASHINGTON',
       'WEST_VIRGINIA', 'WISCONSIN', 'WYOMING', 'DODEA', 'NATIONAL'],
      dtype=object)
```

Ввод [12]:

```
le = LabelEncoder()
state_data_le = le.fit_transform(state_data['STATE'])
```

Ввод [13]:

```
le.classes_
```

Out[13]:

```
array(['ALABAMA', 'ALASKA', 'ARIZONA', 'ARKANSAS', 'CALIFORNIA',
       'COLORADO', 'CONNECTICUT', 'DELAWARE', 'DISTRICT_OF_COLUMBIA',
       'DODEA', 'FLORIDA', 'GEORGIA', 'HAWAII', 'IDAHO', 'ILLINOIS',
       'INDIANA', 'IOWA', 'KANSAS', 'KENTUCKY', 'LOUISIANA', 'MAINE',
       'MARYLAND', 'MASSACHUSETTS', 'MICHIGAN', 'MINNESOTA',
       'MISSISSIPPI', 'MISSOURI', 'MONTANA', 'NATIONAL', 'NEBRASKA',
       'NEVADA', 'NEW_HAMPSHIRE', 'NEW_JERSEY', 'NEW_MEXICO', 'NEW_YORK',
       'NORTH_CAROLINA', 'NORTH_DAKOTA', 'OHIO', 'OKLAHOMA', 'OREGON',
       'PENNSYLVANIA', 'RHODE_ISLAND', 'SOUTH_CAROLINA', 'SOUTH_DAKOTA',
       'TENNESSEE', 'TEXAS', 'UTAH', 'VERMONT', 'VIRGINIA', 'WASHINGTON',
       'WEST_VIRGINIA', 'WISCONSIN', 'WYOMING'], dtype=object)
```

Ввод [14]:

```
state_data_le
```

Out[14]:

```
array([ 0,  1,  2, ..., 50, 51, 52])
```

Каждому штату в датасете присваиваем число, которое его заменит

Ввод [15]:

```
np.unique(state_data_le)
```

Out[15]:

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
       34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
       51, 52])
```

Теперь посредством OneHotEncoding

Ввод [16]:

```
ohe = OneHotEncoder()
state_data_ohe = ohe.fit_transform(state_data[['STATE']])
```

Ввод [17]:

```
state_data.shape
```

Out[17]:

```
(1715, 1)
```

Как видно, один столбец преобразился в 53

Ввод [18]:

```
state_data_ohe.shape
```

Out[18]:

```
(1715, 53)
```

Каждая строка датасета теперь имеет только одну единицу среди всех столбцов STATE

Ввод [19]:

```
state_data_ohe.todense()[0:10]
```

Out[19]:

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

График JointPlot

Ввод [20]:

```
sns.jointplot(x="GRADES_ALL_G", y="TOTAL_REVENUE", data=data)
```

Out[20]:

```
<seaborn.axisgrid.JointGrid at 0x20f331286d0>
```