МГТУ им. Н. Э. Баумана, кафедра ИУ5

курс “Технологии машинного обучения”

Рубежный контроль №1

«Технологии разведочного анализа и обработки данных»

ВЫПОЛНИЛ:

Пученков Д.О.

Группа: ИУ5-61Б

Вариант: 20

ПРОВЕРИЛ:

Гапанюк Ю.Е.

Москва, 2020 г.

Задание (№3):

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?

Набор данных (№4): https://www.kaggle.com/noriuk/us-education-datasets-unification-project (файл states_all.csv)

Выполненная работа:

1. Загрузка и первичный анализ данных

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

```
data = pd.read_csv("../datasets/states_all.csv", sep=',')
```

```
# размер набора данных
data.shape
```

```
(1715, 25)
```

```
# типы колонок
data.dtypes
```

```
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
```

```
# проверим, есть ли пропущенные значения
data.isnull().sum()
```

```
PRIMARY_KEY                       0
STATE                             0
YEAR                              0
ENROLL                          491
TOTAL_REVENUE                   440
FEDERAL_REVENUE                 440
STATE_REVENUE                   440
LOCAL_REVENUE                   440
TOTAL_EXPENDITURE               440
INSTRUCTION_EXPENDITURE         440
SUPPORT_SERVICES_EXPENDITURE    440
OTHER_EXPENDITURE               491
CAPITAL_OUTLAY_EXPENDITURE      440
GRADES_PK_G                     173
GRADES_KG_G                      83
GRADES_4_G                       83
GRADES_8_G                       83
GRADES_12_G                      83
GRADES_1_8_G                    695
GRADES_9_12_G                   644
GRADES_ALL_G                     83
AVG_MATH_4_SCORE                1150
AVG_MATH_8_SCORE                1113
AVG_READING_4_SCORE             1065
AVG_READING_8_SCORE             1153
dtype: int64
```

```
# первые 5 строк
data.head()
```

	PRIMARY_KEY	STATE	YEAR	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVENUE	LOCAL_
0	1992_ALABAMA	ALABAMA	1992	NaN	2678885.0	304177.0	1659028.0	715680.(
1	1992_ALASKA	ALASKA	1992	NaN	1049591.0	106780.0	720711.0	222100.(
2	1992_ARIZONA	ARIZONA	1992	NaN	3258079.0	297888.0	1369815.0	1590376
3	1992_ARKANSAS	ARKANSAS	1992	NaN	1711959.0	178571.0	958785.0	574603.(
4	1992_CALIFORNIA	CALIFORNIA	1992	NaN	26260025.0	2072470.0	16546514.0	7641041

5 rows × 25 columns

2.1. Кодирование категорий целочисленными значениями - label encoding

```
cat_data = np.array(data[['STATE']])
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
cat_enc = pd.DataFrame({'c1':cat_data.T[0]})
le = LabelEncoder()
cat_enc_le = le.fit_transform(cat_enc['c1'])
cat_enc['c1'].unique()
```

```
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)
```

```
row = np.unique(cat_enc_le)
```

```
le.inverse_transform(row)
```

```
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)
```

2.2. Кодирование категорий наборами бинарных значений - one-hot encoding

```
ohe = OneHotEncoder()
cat_enc_ohe = ohe.fit_transform(cat_enc[['c1']])
cat_enc.shape
```

```
(1715, 1)
```

```
cat_enc_ohe.shape
```

```
(1715, 53)
```

```
cat_enc_ohe
```

```
<1715x53 sparse matrix of type '<class 'numpy.float64'>'
        with 1715 stored elements in Compressed Sparse Row format>
```

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

```
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.],
```

```
cat_enc.head(10)
```

	c1
0	ALABAMA
1	ALASKA
2	ARIZONA
3	ARKANSAS
4	CALIFORNIA
5	COLORADO
6	CONNECTICUT
7	DELAWARE
8	DISTRICT_OF_COLUMBIA
9	FLORIDA

2.3. Pandas get_dummies - быстрый вариант one-hot кодирования

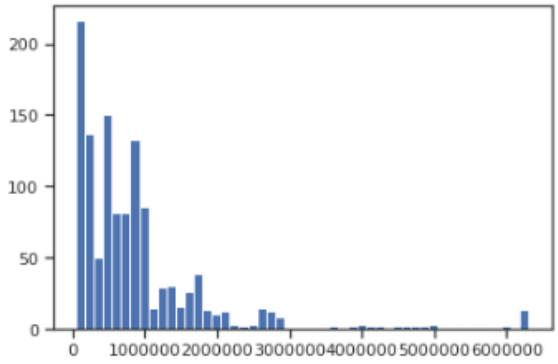
```
pd.get_dummies(cat_enc, dummy_na=True).head()
```

	c1_ALABAMA	c1_ALASKA	c1_ARIZONA	c1_ARKANSAS	c1_CALIFORNIA	c1_COLORADO	c1_CONNECTICUT	c1_D
0	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0
3	0	0	0	1	0	0	0	0
4	0	0	0	0	1	0	0	0

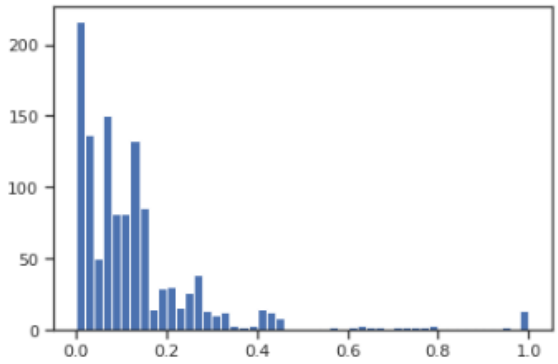
3. Масштабирование данных. MinMax масштабирование

```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normalizer
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data[['ENROLL']])
plt.hist(data['ENROLL'], 50)
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/numpy/lib/histograms.py:839: RuntimeWarning: invalid value encounte
red in greater_equal
  keep = (tmp_a >= first_edge)
/usr/local/lib/python3.7/dist-packages/numpy/lib/histograms.py:840: RuntimeWarning: invalid value encounte
red in less_equal
  keep &= (tmp_a <= last_edge)
```



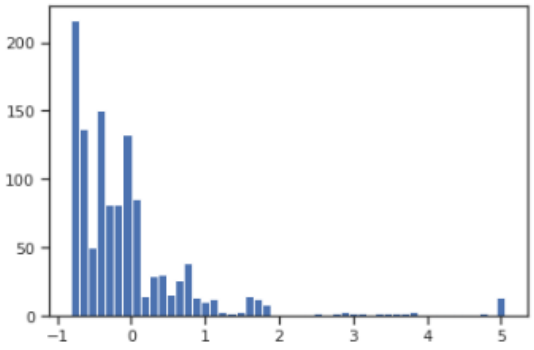

```
plt.hist(sc1_data, 50)
plt.show()
```



3.2. Масштабирование данных на основе Z-оценки - StandardScaler

```
sc2 = StandardScaler()
sc2_data = sc2.fit_transform(data[['ENROLL']])
plt.hist(sc2_data, 50)
plt.show()
```

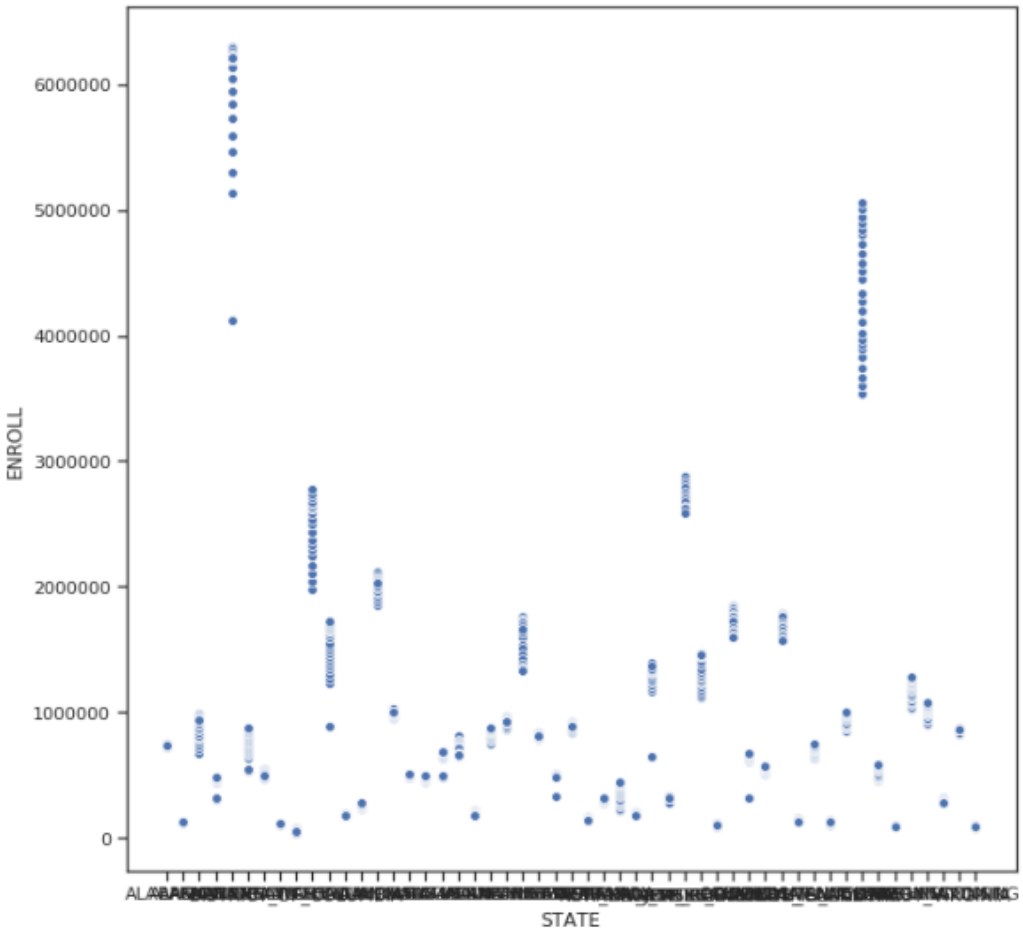
```
/usr/local/lib/python3.7/dist-packages/numpy/lib/histograms.py:839: RuntimeWarning: invalid value encounte
red in greater_equal
  keep = (tmp_a >= first_edge)
/usr/local/lib/python3.7/dist-packages/numpy/lib/histograms.py:840: RuntimeWarning: invalid value encounte
red in less_equal
  keep &= (tmp_a <= last_edge)
```



4. Дополнительное задание по группам: «Диаграмма рассеяния»

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='STATE', y='ENROLL', data=data)
```

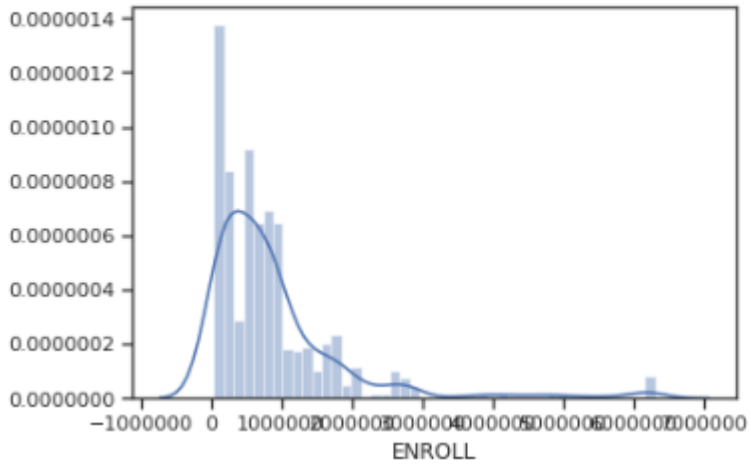
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe44b10610>
```



Т.к. диаграмма рассеяния не дает корректного представления о данных, применим гистограмму:

```
sns.distplot(data['ENROLL'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fbe6e8f43d0>
```



Вывод: В данной работе были применены различные методы обработки данных. Для выбранного датасета лучшим методом масштабирования является MinMax масштабирования. В качестве метода для преобразования категориальных признаков в количественные лучше всего себя показал one-hot encoding.