Рубежный контроль №1

Терентьев В.О. Группа ИУ5-63Б

Вариант 20

Задача. 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?

Дополнительное требование: для произвольной колонки данных построить график "Ящик с усами (boxplot)".

Набор данных: U.S. Education Datasets: Unification Project (https://www.kaggle.com/noriuk/us-education-datasets-unification-project) (файл states_all.csv).

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:

```
# загрузка набора данных
data = pd.read_csv('states_all.csv', sep=",")
# размер набора данных
data.shape
```

Out[2]:

```
(1715, 25)
```

In [3]:

```
# первые 5 строк набора данных
data.head()
```

Out[3]:

	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

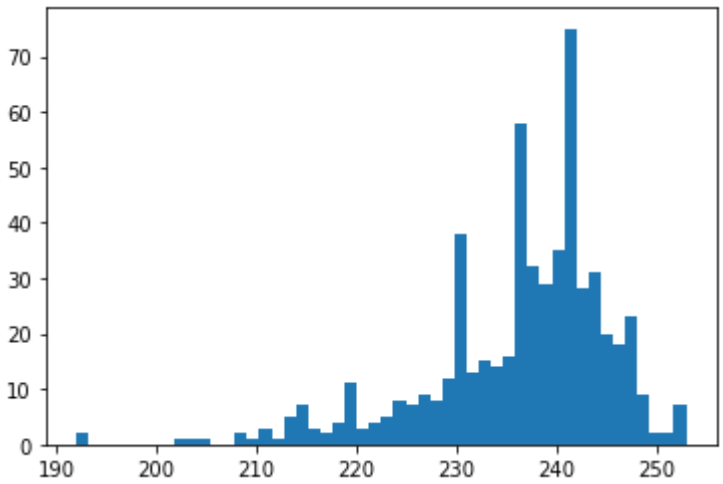
Масштабирование данных:

Для решения этой задачи я буду использовать **MinMax масштабирование**, потому что в случае масштабирования нескольких признаков, лежащих в различных диапазонах, после масштабирования значения всегда будут лежать в одинаковом диапазоне от 0 до 1. Таким образом, данные признаки будут одинаково влиять на модель машинного обучения.

Например, произведем масштабирование признака "AVG_MATH_4_SCORE":

In [4]:

```
# гистограмма распределения данного признака
plt.hist(data['AVG_MATH_4_SCORE'], 50)
plt.show()
```



In [5]:

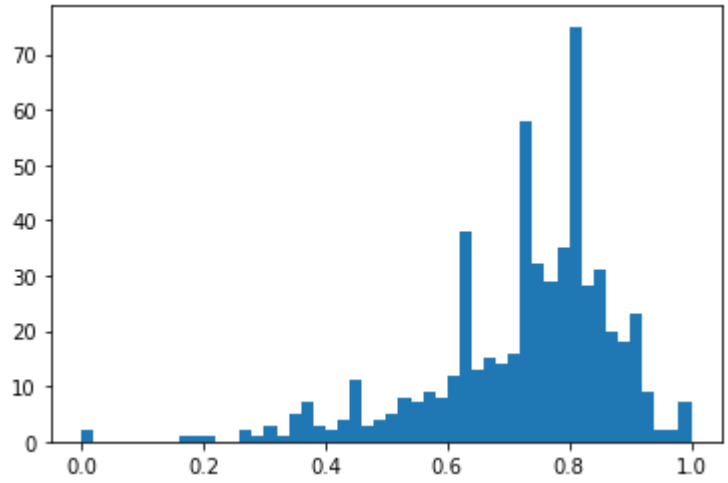
```
from sklearn.preprocessing import MinMaxScaler
```

In [6]:

```
# MinMax масштабирование
mms = MinMaxScaler()
sc_data = mms.fit_transform(data[['AVG_MATH_4_SCORE']])
```

In [7]:

```
# гистограмма распределения после MinMax масштабирования данного признака
plt.hist(sc_data, 50)
plt.show()
```



Преобразование категориальных признаков в количественные:

One-hot encoding:

Я использую **one-hot encoding**, потому что этот метод не задает отношение порядка между значениями данного признака.

Например, выполним преобразование для категориального признака "STATE":

In [8]:

```
# one-hot encoding
pd.get_dummies(data['STATE']).head()
```

Out[8]:

	ALABAMA	ALASKA	ARIZONA	ARKANSAS	CALIFORNIA	COLORADO	CONNECTICUT	DEL
0	1	0	0	0	0	0	0	
1	0	1	0	0	0	0	0	
2	0	0	1	0	0	0	0	
3	0	0	0	1	0	0	0	
4	0	0	0	0	1	0	0	

5 rows × 53 columns

Label encoding:

Этот метод позволяет не расширять признаковое пространство набора данных.

In [9]:

```
from sklearn.preprocessing import LabelEncoder
```

In [10]:

```
# исходные уникальные значения данного признака
data['STATE'].unique()
```

Out[10]:

```
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)
```

In [11]:

```
# label encoding
le = LabelEncoder()
data_le = le.fit_transform(data['STATE'])
```

In [12]:

```
# уникальные значения после label encoding
np.unique(data_le)
```

Out[12]:

```
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])
```

In [13]:

```
# обратное преобразование
le.inverse_transform(data_le)
```

Out[13]:

```
array(['ALABAMA', 'ALASKA', 'ARIZONA', ..., 'WEST_VIRGINIA', 'WISCONSIN',
       'WYOMING'], dtype=object)
```

Построение графика "Ящик с усами (boxplot)":

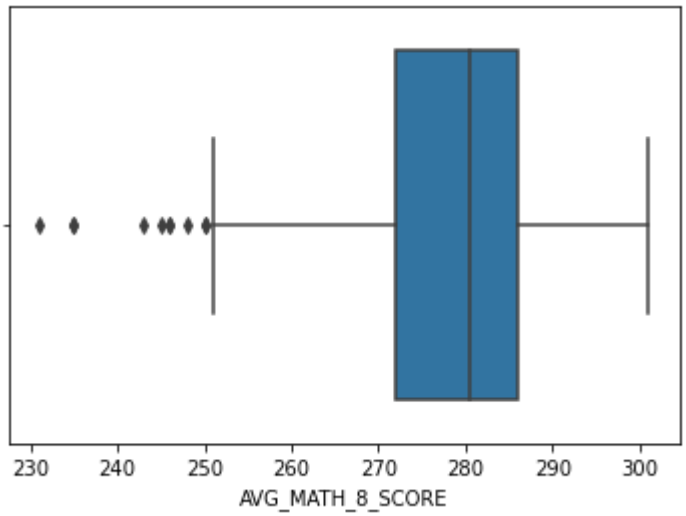
Отображает одномерное распределение вероятности. Построение графика для колонки данных "AVG_MATH_8_SCORE".

In [14]:

```
sns.boxplot(x=data['AVG_MATH_8_SCORE'])
```

Out[14]:

```
<AxesSubplot:xlabel='AVG_MATH_8_SCORE'>
```



In [15]:

```
# по вертикали
sns.boxplot(y=data['AVG_MATH_8_SCORE'])
```

Out[15]:

```
<AxesSubplot:ylabel='AVG_MATH_8_SCORE'>
```