РК1 ИУ5-65Б Нагдимаев Ильягу

Номер варианта - 12

Номер задачи - 2

Номер набора данных, указанного в задаче – 4

Условие задания:

Для заданного набора данных проведите обработку пропусков в данных для одного категориального и одного количественного признака. Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали? Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему?

Дополнительное задание:

Для пары произвольных колонок данных построить график "Парные диаграммы".

Набор данных:

https://www.kaggle.com/noriuk/us-education-datasets-unification-project (файл states_all.csv)

Импорт библиотек

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
import warnings
warnings.filterwarnings('ignore')
sns.set(style="ticks")
%matplotlib inline
```

In [2]:
```
data = pd.read_csv('sample_data/states_all.csv')
```

In [3]:
```
data.head()
```

Out[3]:

	PRIMARY_KEY	STATE	YEAR	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVEN
0	1992_ALABAMA	ALABAMA	1992	NaN	2678885.0	304177.0	16590
1	1992_ALASKA	ALASKA	1992	NaN	1049591.0	106780.0	7207
2	1992_ARIZONA	ARIZONA	1992	NaN	3258079.0	297888.0	13698
3	1992_ARKANSAS	ARKANSAS	1992	NaN	1711959.0	178571.0	9587
4	1992_CALIFORNIA	CALIFORNIA	1992	NaN	26260025.0	2072470.0	165465

In [4]:
```
data.dtypes
```

Out[4]:
```
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
```

In [5]:
```
data.isnull().sum()
# проверим есть ли пропущенные значения
```

Out[5]:
```
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
```

In [6]:
```
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
 3   ENROLL                        1224 non-null   int64
 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)           float64
memory usage: 335.1+ KB
```

Обработка ненужных данных

In [7]:
```
# Удаляем столбцы, которые не несут значимой информации
data.drop(['INSTRUCTION_EXPENDITURE','YEAR'], axis = 1, inplace = True)
```

In [8]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1715 entries, 0 to 1714
Data columns (total 23 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   PRIMARY_KEY                   1715 non-null   object 
 1   STATE                         1715 non-null   object 
 2   ENROLL                        1224 non-null   float64
 3   TOTAL_REVENUE                 1275 non-null   float64
 4   FEDERAL_REVENUE               1275 non-null   float64
 5   STATE_REVENUE                 1275 non-null   float64
 6   LOCAL_REVENUE                 1275 non-null   float64
 7   TOTAL_EXPENDITURE             1275 non-null   float64
 8   SUPPORT_SERVICES_EXPENDITURE  1275 non-null   float64
 9   OTHER_EXPENDITURE             1224 non-null   float64
 10  CAPITAL_OUTLAY_EXPENDITURE    1275 non-null   float64
 11  GRADES_PK_G                   1542 non-null   float64
 12  GRADES_KG_G                   1632 non-null   float64
 13  GRADES_4_G                    1632 non-null   float64
 14  GRADES_8_G                    1632 non-null   float64
 15  GRADES_12_G                   1632 non-null   float64
 16  GRADES_1_8_G                  1020 non-null   float64
 17  GRADES_9_12_G                 1071 non-null   float64
 18  GRADES_ALL_G                  1632 non-null   float64
 19  AVG_MATH_4_SCORE              565 non-null    float64
 20  AVG_MATH_8_SCORE              602 non-null    float64
 21  AVG_READING_4_SCORE           650 non-null    float64
 22  AVG_READING_8_SCORE           562 non-null    float64
dtypes: float64(21), object(2)
memory usage: 308.3+ KB
```

In [9]:
```
# Заполняем отсутствующие значения
```

```
data['TOTAL_REVENUE'] = data['TOTAL_REVENUE'].replace(0,np.nan)
data['TOTAL_REVENUE'] = data['TOTAL_REVENUE'].fillna(data['TOTAL_REVENUE'].mean())
```

In [10]:
```
data.head()
```

Out[10]:

	PRIMARY_KEY	STATE	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVENUE	LO
0	1992_ALABAMA	ALABAMA	NaN	2678885.0	304177.0	1659028.0	
1	1992_ALASKA	ALASKA	NaN	1049591.0	106780.0	720711.0	
2	1992_ARIZONA	ARIZONA	NaN	3258079.0	297888.0	1369815.0	
3	1992_ARKANSAS	ARKANSAS	NaN	1711959.0	178571.0	958785.0	
4	1992_CALIFORNIA	CALIFORNIA	NaN	26260025.0	2072470.0	16546514.0	

5 rows × 23 columns

In [11]:
```
data.isnull().sum()
# проверим есть ли пропущенные значения в столбце business_latitude
```

Out[11]:
```
PRIMARY_KEY                        0
STATE                              0
ENROLL                           491
TOTAL_REVENUE                      0
FEDERAL_REVENUE                  440
STATE_REVENUE                    440
LOCAL_REVENUE                    440
TOTAL_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
```

In [12]:
```
total_count = data.shape[0]
print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 1715
```

Обработка пропусков категориальных данных

In [13]:
```
# Выберем категориальные колонки с пропущенными значениями
# Цикл по колонкам датасета
cat_cols = []
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='object'):
        cat_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {}, {}%.'.forma
```

In [14]:

```
# Заполняем отсутствующие значения
data['STATE'] = data.fillna("Nane")
data.head()
```

Out[14]:

	PRIMARY_KEY	STATE	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVENU
0	1992_ALABAMA	1992_ALABAMA	NaN	2678885.0	304177.0	1659028
1	1992_ALASKA	1992_ALASKA	NaN	1049591.0	106780.0	720711
2	1992_ARIZONA	1992_ARIZONA	NaN	3258079.0	297888.0	1369815
3	1992_ARKANSAS	1992_ARKANSAS	NaN	1711959.0	178571.0	958785
4	1992_CALIFORNIA	1992_CALIFORNIA	NaN	26260025.0	2072470.0	16546514

5 rows × 23 columns

In [15]:

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

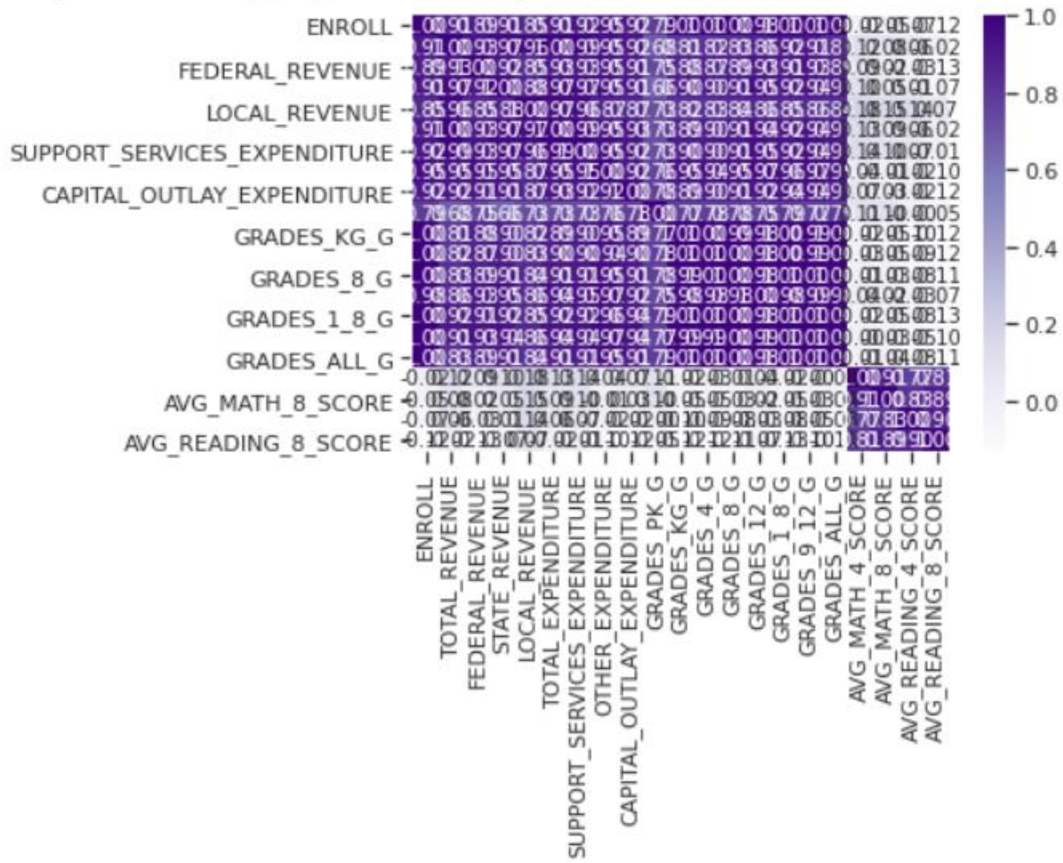
Out[15]:

```
PRIMARY_KEY                        0
STATE                              0
ENROLL                           491
TOTAL_REVENUE                      0
FEDERAL_REVENUE                  440
STATE_REVENUE                    440
LOCAL_REVENUE                    440
TOTAL_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
```

Корреляционный анализ данных

```
[ ] sns.heatmap(data.corr(), cmap = 'Purples', annot = True, fmt = '.3f')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff72e182750>
```



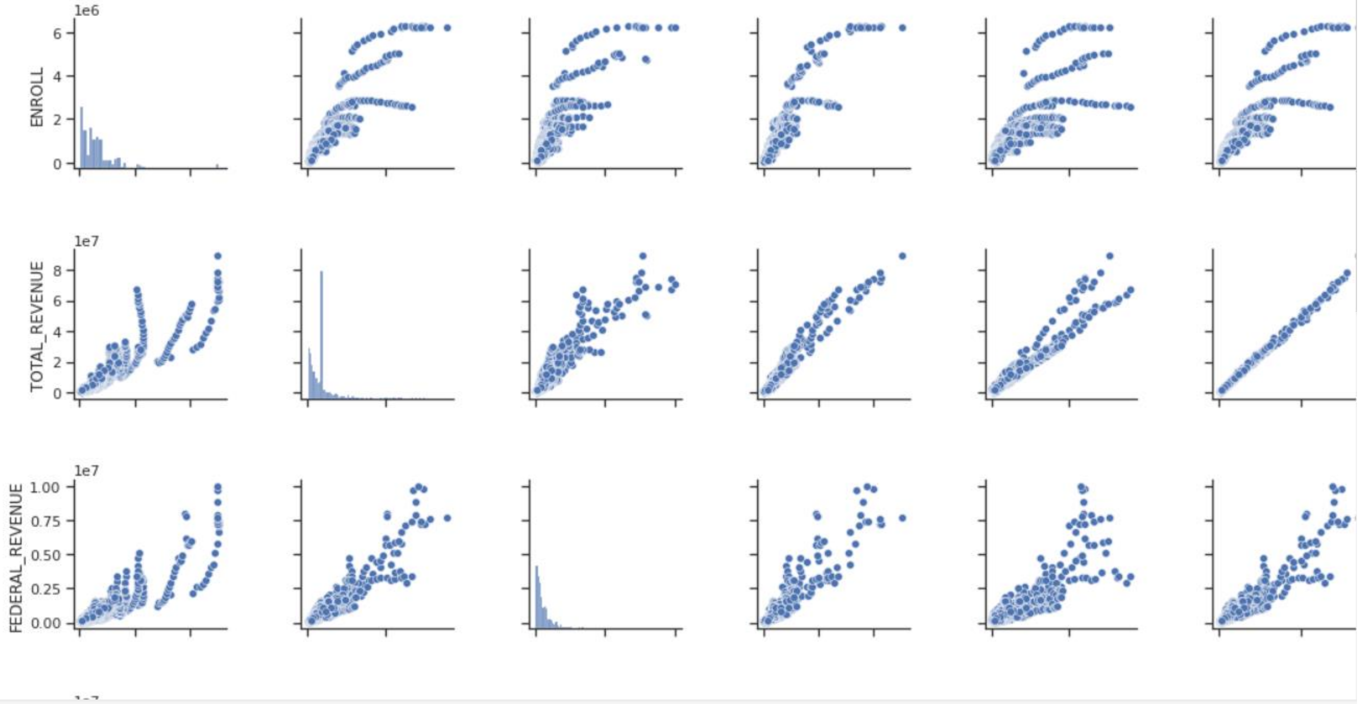
```
data.corr()
```

	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVENUE	LOCAL_REVENUE	TOTAL_EXPENDITURE	SUPPORT_SERVI
ENROLL	1.000000	0.913978	0.893697	0.914379	0.846851	0.914920	
TOTAL_REVENUE	0.913978	1.000000	0.928356	0.972579	0.964968	0.999023	
FEDERAL_REVENUE	0.893697	0.928356	1.000000	0.920708	0.848962	0.928689	
STATE_REVENUE	0.914379	0.972579	0.920708	1.000000	0.880103	0.970049	
LOCAL_REVENUE	0.846851	0.964968	0.848962	0.880103	1.000000	0.965364	
TOTAL_EXPENDITURE	0.914920	0.999023	0.928689	0.970049	0.965364	1.000000	
SUPPORT_SERVICES_EXPENDITURE	0.917475	0.994848	0.931735	0.968800	0.957046	0.993309	
OTHER_EXPENDITURE	0.953018	0.947008	0.947400	0.950481	0.869888	0.946084	
CAPITAL_OUTLAY_EXPENDITURE	0.918076	0.924552	0.907773	0.914920	0.865936	0.932388	
GRADES_PK_G	0.786993	0.682126	0.746690	0.658112	0.729332	0.729214	
GRADES_KG_G	0.995072	0.806880	0.878342	0.898232	0.820106	0.892401	
GRADES_4_G	0.997529	0.816706	0.874619	0.898295	0.827498	0.896040	
GRADES_8_G	0.998371	0.834125	0.887448	0.909776	0.840705	0.909126	
GRADES_12_G	0.983393	0.863705	0.928369	0.945648	0.863032	0.938884	
GRADES_1_8_G	0.999096	0.919819	0.913000	0.919280	0.849337	0.921245	
GRADES_9_12_G	0.997224	0.912582	0.931070	0.940813	0.864700	0.939811	
GRADES_ALL_G	0.998879	0.828885	0.885103	0.907637	0.841279	0.908172	
AVG_MATH_4_SCORE	-0.017301	0.123959	0.090260	0.102318	0.175046	0.134774	

Парные диаграммы

```
[ ] sns.pairplot(data)
```

<seaborn.axisgrid.PairGrid at 0x7ff72e12e390>




```
[ ] sns.pairplot(data, hue = 'STATE_REVENUE')
```

<seaborn.axisgrid.PairGrid at 0x7ff72174a450>