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 по рубежному контролю №1

По дисциплине:
«Технологии машинного обучения»
Вариант 13

Выполнил:

Студент группы ИУ5 _____ **Овчинников С.С.**

(Подпись, дата) (Фамилия И.О.)

Проверил:

_____ **Гапанюк Ю. Е.**

(Подпись, дата) (Фамилия И.О.)

Москва, 2021

Задача №2.

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?
Для пары произвольных колонок данных построить график "Диаграмма рассеяния".

Набор данных

https://www.kaggle.com/noriuk/us-education-datasets-unification-project

Решение

РК ИУ5-61Б Овчинников Степан Вариант 13

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('states_all_extended.csv')
```

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

Out[3]:

	PRIMARY_KEY	STATE	YEAR	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVENUE	LOCAL_REVENUE	TOTAL_EXPENDITURE	INS
0	1992_ALABAMA	ALABAMA	1992	NaN	2678885.0	304177.0	1659028.0	715680.0	2653798.0	
1	1992_ALASKA	ALASKA	1992	NaN	1049591.0	106780.0	720711.0	222100.0	972488.0	
2	1992_ARIZONA	ARIZONA	1992	NaN	3258079.0	297888.0	1369815.0	1590376.0	3401580.0	
3	1992_ARKANSAS	ARKANSAS	1992	NaN	1711959.0	178571.0	958785.0	574603.0	1743022.0	
4	1992_CALIFORNIA	CALIFORNIA	1992	NaN	26260025.0	2072470.0	16546514.0	7641041.0	27138832.0	

5 rows × 266 columns

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

Out[4]:
```
PRIMARY_KEY             object
STATE                   object
YEAR                     int64
ENROLL                 float64
TOTAL_REVENUE          float64
                        ...
G08_AM_A_MATHEMATICS   float64
G08_HP_A_READING       float64
G08_HP_A_MATHEMATICS   float64
G08_TR_A_READING       float64
G08_TR_A_MATHEMATICS   float64
Length: 266, 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
                       ...
G08_AM_A_MATHEMATICS   1655
G08_HP_A_READING       1701
G08_HP_A_MATHEMATICS   1702
G08_TR_A_READING       1574
G08_TR_A_MATHEMATICS   1570
Length: 266, dtype: int64
```

In [6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1715 entries, 0 to 1714
Columns: 266 entries, PRIMARY_KEY to G08_TR_A_MATHEMATICS
dtypes: float64(263), int64(1), object(2)
memory usage: 3.5+ MB
```

Обработка пропусков

In [7]:
```
# Удаляем столбцы, которые не несут значимой информации
data.drop(['G08_AM_A_MATHEMATICS','G08_HP_A_READING'], axis = 1, inplace = True)
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1715 entries, 0 to 1714
Columns: 264 entries, PRIMARY_KEY to G08_TR_A_MATHEMATICS
dtypes: float64(261), int64(1), object(2)
memory usage: 3.5+ MB
```

Обработка пропусков в числовых данных

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	YEAR	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVENUE	LOCAL_REVENUE	TOTAL_EXPENDITURE	IN$
0	1992_ALABAMA	ALABAMA	1992	NaN	2678885.0	304177.0	1659028.0	715680.0	2653798.0	
1	1992_ALASKA	ALASKA	1992	NaN	1049591.0	106780.0	720711.0	222100.0	972488.0	
2	1992_ARIZONA	ARIZONA	1992	NaN	3258079.0	297888.0	1369815.0	1590376.0	3401580.0	
3	1992_ARKANSAS	ARKANSAS	1992	NaN	1711959.0	178571.0	958785.0	574603.0	1743022.0	
4	1992_CALIFORNIA	CALIFORNIA	1992	NaN	26260025.0	2072470.0	16546514.0	7641041.0	27138832.0	

5 rows × 264 columns

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

Out[11]:
```
PRIMARY_KEY                0
STATE                      0
YEAR                       0
ENROLL                   491
TOTAL_REVENUE              0
                        ...
G08_AS_A_MATHEMATICS    1558
G08_AM_A_READING        1654
G08_HP_A_MATHEMATICS    1702
G08_TR_A_READING        1574
G08_TR_A_MATHEMATICS    1570
Length: 264, dtype: int64
```

Обработка пропусков в категориальных данных

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

```
Всего строк: 1715
```

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 {}, {}%.'.format(col, dt, temp_null_count, temp_perc))
```

In [14]:
```
# Заполняем отсутствующие значения
data['ENROLL'] = data.fillna("Name")
data.head()
```

Out[14]:

	PRIMARY_KEY	STATE	YEAR	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVENUE	LOCAL_REVENUE	TOTAL_EXPENDITI
0	1992_ALABAMA	ALABAMA	1992	1992_ALABAMA	2678885.0	304177.0	1659028.0	715680.0	26537
1	1992_ALASKA	ALASKA	1992	1992_ALASKA	1049591.0	106780.0	720711.0	222100.0	9724
2	1992_ARIZONA	ARIZONA	1992	1992_ARIZONA	3258079.0	297888.0	1369815.0	1590376.0	34015
3	1992_ARKANSAS	ARKANSAS	1992	1992_ARKANSAS	1711959.0	178571.0	958785.0	574603.0	17430
4	1992_CALIFORNIA	CALIFORNIA	1992	1992_CALIFORNIA	26260025.0	2072470.0	16546514.0	7641041.0	271388

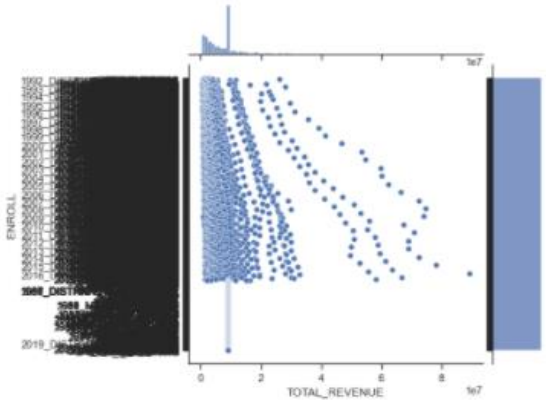
5 rows × 264 columns

In [15]:
```
data.isnull().sum()
# проверим есть ли пропущенные значения в столбце ENROLL
```

Out[15]:
```
PRIMARY_KEY                0
STATE                      0
YEAR                       0
ENROLL                     0
TOTAL_REVENUE              0
                        ...
G08_AS_A_MATHEMATICS    1558
G08_AM_A_READING        1654
G08_HP_A_MATHEMATICS    1702
G08_TR_A_READING        1574
G08_TR_A_MATHEMATICS    1570
Length: 264, dtype: int64
```

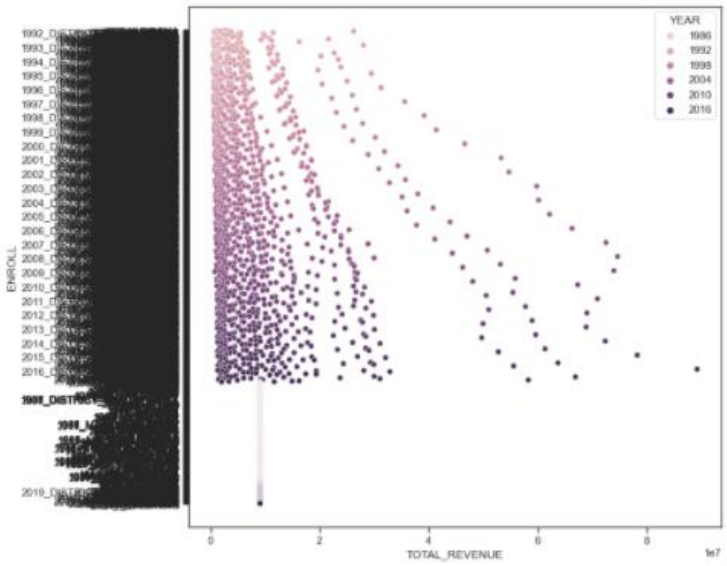
In [16]:
```
# Увеличенные диаграммы рассеяния
sns.jointplot(x = "TOTAL_REVENUE", y = "ENROLL", kind="scatter", data = data)
```

Out[16]: <seaborn.axisgrid.JointGrid at 0x25750cca6a0>



In [17]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x = "TOTAL_REVENUE", y = "ENROLL", data=data, hue='YEAR')
```

Out[17]: <AxesSubplot:xlabel='TOTAL_REVENUE', ylabel='ENROLL'>

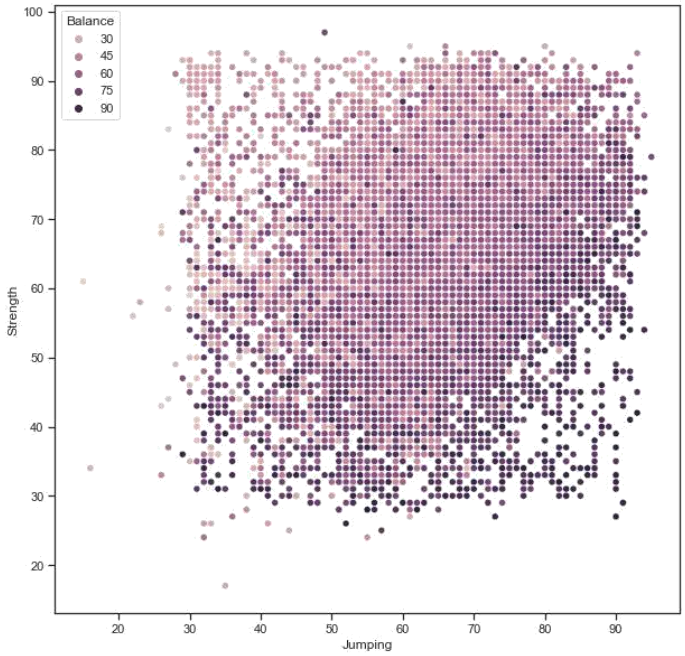


Ввод [6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18207 entries, 0 to 18206
Data columns (total 89 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Unnamed: 0                18207 non-null  int64
 1   ID                        18207 non-null  int64
 2   Name                      18207 non-null  object
 3   Age                       18207 non-null  int64
 4   Photo                     18207 non-null  object
 5   Nationality               18207 non-null  object
 6   Flag                      18207 non-null  object
 7   Overall                   18207 non-null  int64
 8   Potential                 18207 non-null  int64
 9   Club                      17966 non-null  object
 10  Club Logo                 18207 non-null  object
 11  Value                     18207 non-null  object
 12  Wage                      18207 non-null  object
 13  Special                   18207 non-null  int64
 14  Preferred Foot            18159 non-null  object
 15  International Reputation  18159 non-null  float64
 16  Weak Foot                 18159 non-null  float64
 17  Skill Moves               18159 non-null  float64
 18  Work Rate                 18159 non-null  object
 19  Body Type                 18159 non-null  object
 20  Real Face                 18159 non-null  object
 21  Position                  18147 non-null  object
 22  Jersey Number             18147 non-null  float64
 23  Joined                    16654 non-null  object
 24  Loaned From               1264 non-null   object
 25  Contract Valid Until      17918 non-null  object
 26  Height                    18159 non-null  object
 27  Weight                    18159 non-null  object
 28  LS                        16122 non-null  object
 29  ST                        16122 non-null  object
 30  RS                        16122 non-null  object
 31  LW                        16122 non-null  object
 32  LF                        16122 non-null  object
 33  CF                        16122 non-null  object
 32  LF                        16122 non-null  object
 33  CF                        16122 non-null  object
 34  RF                        16122 non-null  object
 35  RW                        16122 non-null  object
 36  LAM                       16122 non-null  object
 37  CAM                       16122 non-null  object
 38  RAM                       16122 non-null  object
 39  LM                        16122 non-null  object
 40  LCM                       16122 non-null  object
 41  CM                        16122 non-null  object
 42  RCM                       16122 non-null  object
 43  RM                        16122 non-null  object
 44  LWB                       16122 non-null  object
 45  LDM                       16122 non-null  object
 46  CDM                       16122 non-null  object
 47  RDM                       16122 non-null  object
 48  RWB                       16122 non-null  object
 49  LB                        16122 non-null  object
 50  LCB                       16122 non-null  object
 51  CB                        16122 non-null  object
 52  RCB                       16122 non-null  object
 53  RB                        16122 non-null  object
 54  Crossing                  18159 non-null  float64
 55  Finishing                 18159 non-null  float64
 56  HeadingAccuracy           18159 non-null  float64
 57  ShortPassing              18159 non-null  float64
 58  Volleys                   18159 non-null  float64
 59  Dribbling                 18159 non-null  float64
 60  Curve                     18159 non-null  float64
 61  FKAccuracy                18159 non-null  float64
 62  LongPassing               18159 non-null  float64
 63  BallControl               18159 non-null  float64
 64  Acceleration              18159 non-null  float64
 65  SprintSpeed               18159 non-null  float64
 66  Agility                   18159 non-null  float64
 67  Reactions                 18159 non-null  float64
 68  Balance                   18159 non-null  float64
 69  ShotPower                 18159 non-null  float64
 70  Jumping                   18159 non-null  float64
 71  Stamina                   18159 non-null  float64
 72  Strength                  18159 non-null  float64
 73  LongShots                 18159 non-null  float64
 74  Aggression                18159 non-null  float64
 75  Interceptions             18159 non-null  float64
 76  Positioning               18159 non-null  float64
 77  Vision                    18159 non-null  float64
```

Ввод [20]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x = "Jumping", y = "Strength", data=data, hue='Balance')
```

Out[20]: <AxesSubplot:xlabel='Jumping', ylabel='Strength'>



Ввод [ ]:
```
# The end.
```