Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

Дисциплина «Технологии машинного обучения»

Отчёт

по рубежному контролю №1

Тема: «Технологии разведочного анализа и обработки данных.»

Вариант 12

Студент:

Крюков Г. М.

Группа ИУ5-61Б

Преподаватель:

Гапанюк Ю. Е.

Москва, 2020 г.

Задание

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?

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

Дополнительные требования по группам:

Для студентов групп ИУ5-61Б, ИУ5Ц-81Б - для пары произвольных колонок данных построить график "Диаграмма рассеяния".

Выполнение задания

```
import numpy as np
import pandas as pd
pd.set_option('display.max.columns', 100)
# to draw pictures in jupyter notebook
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
# we don't like warnings
# you can comment the following 2 lines if you'd like to
import warnings
warnings.filterwarnings('ignore')
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
```

```
data = pd.read_csv('/content/datasets_95317_1078789_states_all.csv')
data.head()
```

	PRIMARY_KEY	STATE	YEAR	ENROLL	TOTAL_REVENUE	FEDERAL_REVENUE	STATE_REVE
0	1992_ALABAMA	ALABAMA	1992	NaN	2678885.0	304177.0	165902
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

```
#посмотрим на 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 значени
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
```

Как мы можем видеть, в датасете нет пропусков категориальных значений.

```
#размер df
total_count = data.shape[0]
```

```
# Выберем числовые колонки с пропущенными значениями
num_cols = []
total_count = data.shape[0]
for col in data.columns:
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='float64' or dt=='int64'):
        num_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))
```

```
Колонка ENROLL. Тип данных float64. Количество пустых значений 491, 28.63%.
Колонка TOTAL_REVENUE. Тип данных float64. Количество пустых значений 440, 25.66%.
Колонка FEDERAL_REVENUE. Тип данных float64. Количество пустых значений 440, 25.66%.
Колонка STATE_REVENUE. Тип данных float64. Количество пустых значений 440, 25.66%.
Колонка LOCAL_REVENUE. Тип данных float64. Количество пустых значений 440, 25.66%.
Колонка TOTAL_EXPENDITURE. Тип данных float64. Количество пустых значений 440, 25.66%.
Колонка INSTRUCTION_EXPENDITURE. Тип данных float64. Количество пустых значений 440, 25.66%.
Колонка SUPPORT_SERVICES_EXPENDITURE. Тип данных float64. Количество пустых значений 440, 25.66%.
Колонка OTHER_EXPENDITURE. Тип данных float64. Количество пустых значений 491, 28.63%.
Колонка CAPITAL_OUTLAY_EXPENDITURE. Тип данных float64. Количество пустых значений 440, 25.66%.
Колонка GRADES_PK_G. Тип данных float64. Количество пустых значений 173, 10.09%.
Колонка GRADES_KG_G. Тип данных float64. Количество пустых значений 83, 4.84%.
Колонка GRADES_4_G. Тип данных float64. Количество пустых значений 83, 4.84%.
Колонка GRADES_8_G. Тип данных float64. Количество пустых значений 83, 4.84%.
Колонка GRADES_12_G. Тип данных float64. Количество пустых значений 83, 4.84%.
Колонка GRADES_1_8_G. Тип данных float64. Количество пустых значений 695, 40.52%.
Колонка GRADES_9_12_G. Тип данных float64. Количество пустых значений 644, 37.55%.
Колонка GRADES_ALL_G. Тип данных float64. Количество пустых значений 83, 4.84%.
Колонка AVG_MATH_4_SCORE. Тип данных float64. Количество пустых значений 1150, 67.06%.
Колонка AVG_MATH_8_SCORE. Тип данных float64. Количество пустых значений 1113, 64.9%.
Колонка AVG_READING_4_SCORE. Тип данных float64. Количество пустых значений 1065, 62.1%.
Колонка AVG_READING_8_SCORE. Тип данных float64. Количество пустых значений 1153, 67.23%.
```

```
#возьмем колонку Enroll
# Запоминаем индексы строк с пустыми значениями
flt_index = data[data['ENROLL'].isnull()].index
flt_index
```

```
Int64Index([   0,    1,    2,    3,    4,    5,    6,    7,    8,    9,
            ...
            1705, 1706, 1707, 1708, 1709, 1710, 1711, 1712, 1713, 1714],
           dtype='int64', length=491)
```

```
data_enroll = data[num_cols][['ENROLL']]
data_enroll.head()
```

	ENROLL
0	NaN
1	NaN
2	NaN
3	NaN
4	NaN

```
# Фильтр для проверки заполнения пустых значений
indicator = MissingIndicator()
mask_missing_values_only = indicator.fit_transform(data_enrol
mask_missing_values_only
```

```
array([[ True],
       [ True],
       [ True],
       ...,
       [ True],
       [ True],
       [ True]])
```

```
strategy='mean'
```

```
def test_num_impute(strategy_param):
    imp_num = SimpleImputer(strategy=strategy_param)
    data_num_imp = imp_num.fit_transform(data_enroll)
    return data_num_imp[mask_missing_values_only]
```

```
new_enroll = pd.DataFrame({'id': flt_index,
                           'ENROLL':test_num_impute('mean')})
new_enroll
```

	id	ENROLL
0	0	917541.566176
1	1	917541.566176
2	2	917541.566176
3	3	917541.566176
4	4	917541.566176
...	...	...
486	1710	917541.566176
487	1711	917541.566176
488	1712	917541.566176
489	1713	917541.566176
490	1714	917541.566176

491 rows × 2 columns

```
for index, row in new_enroll.iterrows():
    data.loc[row['id'], 'ENROLL'] = row['ENROLL']
data
#очистили данные для колонки ENROLL
```

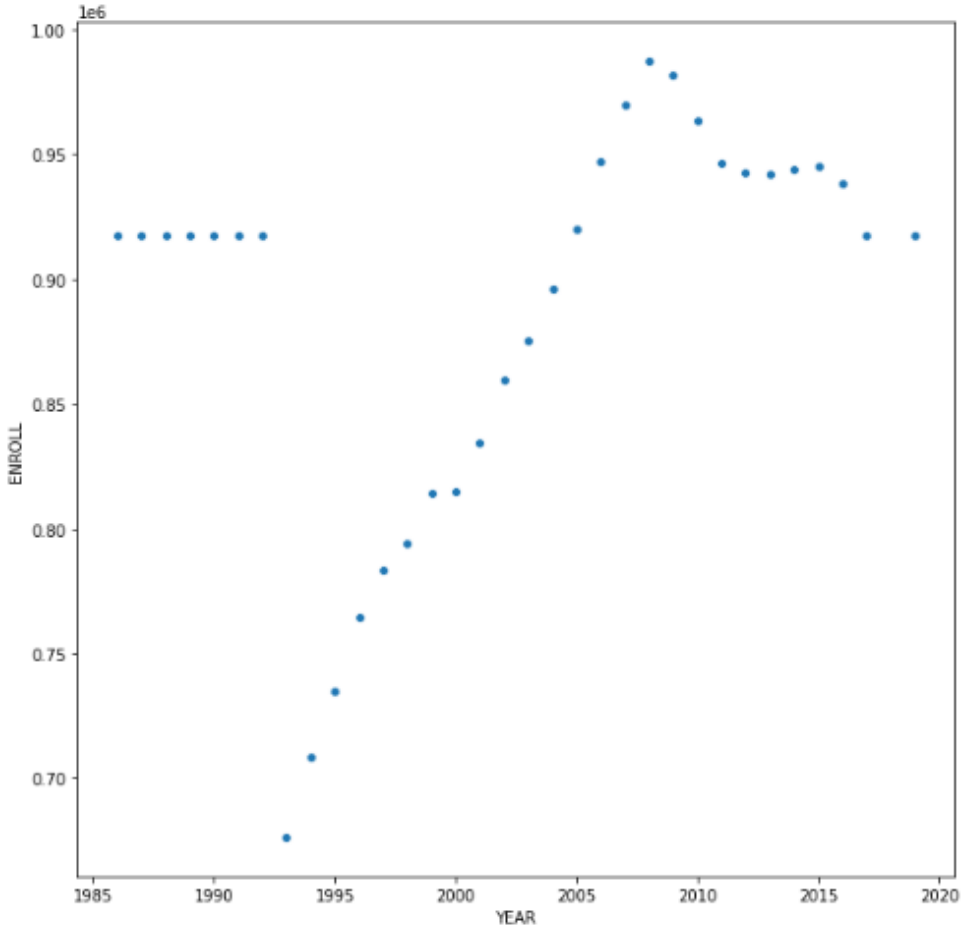
	PRIMARY_KEY	STATE	YEAR	ENROLL	TOTAL_REVENUE	F
0	1992_ALABAMA	ALABAMA	1992	917541.566176	2678885.0	
1	1992_ALASKA	ALASKA	1992	917541.566176	1049591.0	
2	1992_ARIZONA	ARIZONA	1992	917541.566176	3258079.0	
3	1992_ARKANSAS	ARKANSAS	1992	917541.566176	1711959.0	
4	1992_CALIFORNIA	CALIFORNIA	1992	917541.566176	26260025.0	
...	...	...	...	...	...	
1710	2019_VIRGINIA	VIRGINIA	2019	917541.566176	NaN	
1711	2019_WASHINGTON	WASHINGTON	2019	917541.566176	NaN	
1712	2019_WEST_VIRGINIA	WEST_VIRGINIA	2019	917541.566176	NaN	
1713	2019_WISCONSIN	WISCONSIN	2019	917541.566176	NaN	
1714	2019_WYOMING	WYOMING	2019	917541.566176	NaN	

1715 rows × 25 columns

```
# диаграмма рассеяния для первого датасета для штата Аризоны
# зависимость года поступления и числа поступающих

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

```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff1cc796710>
```



Вывод

Таким образом для обработки пропусков в данных для количественного признака использовался метод импьютации средними значениями. Для дальнейшего построения моделей можно использовать все столбцы, обработав пропуски, в зависимости от нужд исследований.