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 ИУ5 «Системы обработки информации и управления»

Отчёт по РК1

«Технологии машинного обучения»

Вариант 12

Выполнила:

студентка группы ИУ5-63Б

Румак Д.П.

Преподаватель:

Гапанюк Ю. Е.

2023 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?

Для студентов групп ИУ5-63Б - для произвольной колонки данных построить график "Ящик с усами (boxplot)".

https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

Решение:

Загружаем датасет и подключаем необходимые библиотеки:

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

```
[2]  data = pd.read_csv(r"heart.csv")
```

Получим информацию о датасете:

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

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 3   trestbps  1025 non-null   int64  
 4   chol      1025 non-null   int64  
 5   fbs       1025 non-null   int64  
 6   restecg   1025 non-null   int64  
 7   thalach   1025 non-null   int64  
 8   exang     1025 non-null   int64  
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64  
 11  ca        1025 non-null   int64  
 12  thal      1025 non-null   int64  
 13  target    1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

Данный датасет не имеет пропусков поэтому заменим его на другой.

```
[10] data=pd.read_csv(r"uk_universities.csv")
```

```
[11] data.head()
```

	Название университета	Регион	Год основания	Девиз	Национальный ранг	Мировой рейтинг	Оценка мировых рейтингов	Minimum_IELTS_score	Иностранные студенты	Ou студе
0	University of Cambridge	East of England	1209	From here, light and sacred draughts	1	4	94.1	6.5	20.20%	85
1	University of Oxford	South East England	1096	The Lord is my light	2	2	93.3	6.5	16.80%	86
2	University of St Andrews	Scotland	1413	Ever to excel	3	86	75.8	6.5	40.40%	87
3	Imperial College London	London	1907	Knowledge is the adornment and safeguard of th...	4	8	86.6	6.5	41.40%	77
4	Loughborough University	East Midlands	1966	With Truth, Knowledge and Labour	5	404	72.8	5.5	22.00%	85

```
[12] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 126 entries, 0 to 125
Data columns (total 17 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   Название университета                 126 non-null    object 
 1   Регион                                126 non-null    object 
 2   Год основания                         126 non-null    int64  
 3   Девиз                                 112 non-null    object 
 4   Национальный ранг                     126 non-null    int64  
 5   Мировой рейтинг                       126 non-null    int64  
 6   Оценка мировых рейтингов              82 non-null     float64
 7   Minimum_IELTS_score                   126 non-null    float64
 8   Иностранные студенты                  126 non-null    object 
 9   Оценка студентов                      126 non-null    object 
 10  Кол-во поступивших студентов (тыс.)   126 non-null    int64  
 11  Кол-во преподавательского состава     126 non-null    float64
 12  Тип управления университета           126 non-null    object 
 13  Местоположение кампуса                109 non-null    object 
 14  Стоимость жизни в год                 126 non-null    int64  
 15  Широта                                126 non-null    float64
 16  Долгота                               126 non-null    object 
dtypes: float64(4), int64(5), object(8)
memory usage: 16.9+ KB
```

```
data.isnull().sum()
```

```
Название университета                  0
Регион                                 0
Год основания                          0
Девиз                                 14
Национальный ранг                      0
Мировой рейтинг                        0
Оценка мировых рейтингов              44
Minimum_IELTS_score                    0
Иностранные студенты                   0
Оценка студентов                       0
Кол-во поступивших студентов (тыс.)    0
Кол-во преподавательского состава      0
Тип управления университета            0
Местоположение кампуса                17
Стоимость жизни в год                  0
Широта                                 0
Долгота                                0
dtype: int64
```

Категориальные признаки: "Девиз", "Местоположение кампуса".
Количественный признак: "Оценка мировых рейтингов".

Выберем колонки с категориальными признаками, которые содержат пропущенные значения

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

    Всего строк: 126
```

```
[7] from sklearn.impute import SimpleImputer
    from sklearn.impute import MissingIndicator
```

```
# 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))
```

```
Колонка Девиз. Тип данных object. Количество пустых значений 14, 11.11%.
Колонка Местоположение кампуса. Тип данных object. Количество пустых значений 17, 13.49%.
```

Будем работать с признаком "Местоположение кампуса".

Проверим уникальные значения.

```
[13] cat_temp_data = data[['Местоположение кампуса']]
     cat_temp_data.head()
```

	Местоположение кампуса
0	Urban
1	Urban
2	Suburban
3	Urban
4	Suburban

```
[15] cat_temp_data['Местоположение кампуса'].unique()

     array(['Urban', 'Suburban', 'Rural', nan], dtype=object)
```

Так как мы не можем точно определить местоположение кампуса университетов, где стоят пропуски, просто напишем, что данные не указаны. Заменим пропуски константой "NotIndicated":

```
imp3 = SimpleImputer(missing_values=np.nan, strategy='constant', fill_value='NotIndicated')
data_imp3 = imp3.fit_transform(cat_temp_data)
data_imp3
```

```
       ['Urban'],
       ['NotIndicated'],
       ['Urban'],
       ['NotIndicated'],
       ['Suburban'],
       ['Urban'],
       ['Urban'],
       ['NotIndicated'],
```

```
np.unique(data_imp3)

array(['NotIndicated', 'Rural', 'Suburban', 'Urban'], dtype=object)
```

```
data_imp3[data_imp3=='NotIndicated'].size
```

```
17
```

Как мы можем увидеть, пропуски отсутствуют.

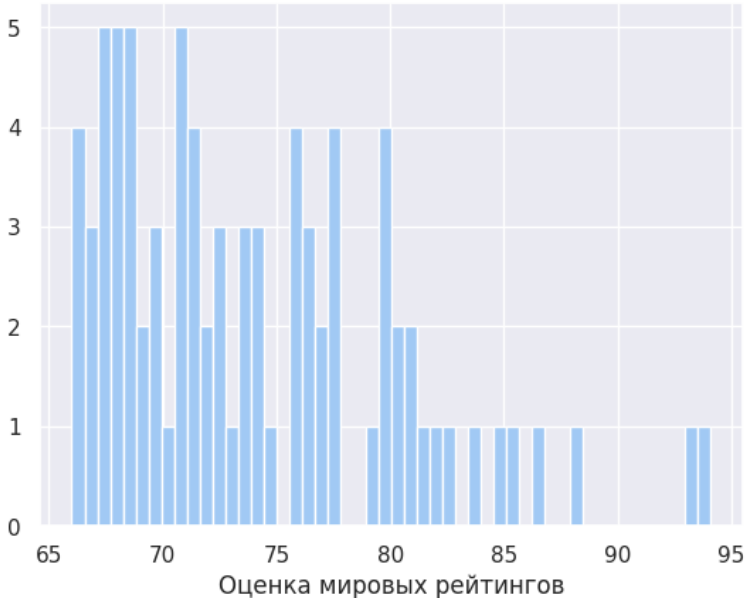
Проведем обработку данных с числовым признаком "Оценка мировых рейтингов".

```
[16] # Фильтр по колонкам с пропущенными значениями
     data_num = data[['Оценка мировых рейтингов']]
     data_num
```

	Оценка мировых рейтингов
0	94.1
1	93.3
2	75.8
3	86.6
4	72.8
...	...
121	NaN
122	NaN
123	NaN
124	NaN
125	NaN

126 rows × 1 columns

```
plt.hist(data_num['Оценка мировых рейтингов'], 50)
plt.xlabel('Оценка мировых рейтингов')
plt.show()
```



```
data_num.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Оценка мировых рейтингов	82.0	74.071951	6.364645	66.0	68.725	72.6	77.7	94.1

```
[18] indicator = MissingIndicator()
     mask_missing_values_only = indicator.fit_transform(data_num)
     mask_missing_values_only
```

```
            [False],
            [False],
            [False],
            [False],
            [False],
            [False],
```

```
[19] strategies=['mean', 'median', 'most_frequent']
```

```
[20] # Более сложная функция, которая позволяет задавать колонку и вид импьютации
     def test_num_impute_col(dataset, column, strategy_param):
         temp_data = dataset[[column]]

         indicator = MissingIndicator()
         mask_missing_values_only = indicator.fit_transform(temp_data)

         imp_num = SimpleImputer(strategy=strategy_param)
         data_num_imp = imp_num.fit_transform(temp_data)

         filled_data = data_num_imp[mask_missing_values_only]

         return column, strategy_param, filled_data.size, filled_data[0], filled_data[filled_data.size-1]
```

```
[21] test_num_impute_col(data_num, 'Оценка мировых рейтингов', strategies[0])
```

```
('Оценка мировых рейтингов', 'mean', 44, 74.07195121951219, 74.07195121951219)
```

```
[22] test_num_impute_col(data_num, 'Оценка мировых рейтингов', strategies[1])
```

```
('Оценка мировых рейтингов', 'median', 44, 72.6, 72.6)
```

```
[23] test_num_impute_col(data_num, 'Оценка мировых рейтингов', strategies[2])
```

```
('Оценка мировых рейтингов', 'most_frequent', 44, 71.1, 71.1)
```

Заполним пропуски медианой, так как при таком заполнении сохраняется распределение значений:

```
data['Оценка мировых рейтингов'] = data['Оценка мировых рейтингов'].fillna(data['Оценка мировых рейтингов'].median())
```

При заполнении пропусков была использована импьютация константным и медианным значениями.

Вернемся к первому набору данных. Целевая переменная - "target" (наличие или отсутствие заболевания пациента).

Чтобы ответить на вопрос, какие признаки лучше всего использовать для построения модели машинного обучения, необходимо провести корреляционный анализ.

```
corr = data.corr()
corr.style.background_gradient(cmap="coolwarm")
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1.000000	-0.103240	-0.071966	0.271121	0.219823	0.121243	-0.132696	-0.390227	0.088163	0.208137	-0.169105	0.271551	0.072297	-0.229324
sex	-0.103240	1.000000	-0.041119	-0.078974	-0.198258	0.027200	-0.055117	-0.049365	0.139157	0.084687	-0.026666	0.111729	0.198424	-0.279501
cp	-0.071966	-0.041119	1.000000	0.038177	-0.081641	0.079294	0.043581	0.306839	-0.401513	-0.174733	0.131633	-0.176206	-0.163341	0.434854
trestbps	0.271121	-0.078974	0.038177	1.000000	0.127977	0.181767	-0.123794	-0.039264	0.061197	0.187434	-0.120445	0.104554	0.059276	-0.138772
chol	0.219823	-0.198258	-0.081641	0.127977	1.000000	0.026917	-0.147410	-0.021772	0.067382	0.064880	-0.014248	0.074259	0.100244	-0.099966
fbs	0.121243	0.027200	0.079294	0.181767	0.026917	1.000000	-0.104051	-0.008866	0.049261	0.010859	-0.061902	0.137156	-0.042177	-0.041164
restecg	-0.132696	-0.055117	0.043581	-0.123794	-0.147410	-0.104051	1.000000	0.048411	-0.065606	-0.050114	0.086086	-0.078072	-0.020504	0.134468
thalach	-0.390227	-0.049365	0.306839	-0.039264	-0.021772	-0.008866	0.048411	1.000000	-0.380281	-0.349796	0.395308	-0.207888	-0.098068	0.422895
exang	0.088163	0.139157	-0.401513	0.061197	0.067382	0.049261	-0.065606	-0.380281	1.000000	0.310844	-0.267335	0.107849	0.197201	-0.438029
oldpeak	0.208137	0.084687	-0.174733	0.187434	0.064880	0.010859	-0.050114	-0.349796	0.310844	1.000000	-0.575189	0.221816	0.202672	-0.438441
slope	-0.169105	-0.026666	0.131633	-0.120445	-0.014248	-0.061902	0.086086	0.395308	-0.267335	-0.575189	1.000000	-0.073440	-0.094090	0.345512
ca	0.271551	0.111729	-0.176206	0.104554	0.074259	0.137156	-0.078072	-0.207888	0.107849	0.221816	-0.073440	1.000000	0.149014	-0.382085
thal	0.072297	0.198424	-0.163341	0.059276	0.100244	-0.042177	-0.020504	-0.098068	0.197201	0.202672	-0.094090	0.149014	1.000000	-0.337838
target	-0.229324	-0.279501	0.434854	-0.138772	-0.099966	-0.041164	0.134468	0.422895	-0.438029	-0.438441	0.345512	-0.382085	-0.337838	1.000000

Проанализировав матрицу, можем сказать, что все признаки слабо коррелируют друг с другом, в том числе и наша целевая переменная, поэтому построение модели по данной выборке нецелесообразно.

Дополнительное задание.

Построение графика "Ящик с усами (boxplot)"

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

```
<Axes: xlabel='Национальный ранг'>
```