

Московский государственный технический университет им. Н.Э. Баумана  
Факультет «Информатика и системы управления»  
Кафедра «Автоматизированные системы обработки информации и  
управления»



**Отчет**  
**Рулежный контроль № 1**  
**По курсу «Технологии машинного обучения»**

**ИСПОЛНИТЕЛЬ:**

Группа ИУ5-65Б

Голубев С.Н.

"17" апреля 2020 г.

**ПРЕПОДАВАТЕЛЬ:**

Гапанюк Ю.Е.

\_\_\_\_\_

"\_\_" \_\_\_\_\_ 2021 г.

Задача №1.

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?

Дополнение для ИУ5-65Б

Для студентов группы ИУ5-65Б - для набора данных построить "парные диаграммы".

Наборы данных: <https://www.kaggle.com/ronitf/heart-disease-uci>

```
B [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

```
B [2]: # Загрузим выборку данных
data = pd.read_csv('heart.csv', sep=",")
```

```
B [3]: # размер набора данных
data.shape
```

```
Out[3]: (303, 14)
```

```
B [4]: # типы колонок
data.dtypes
```

```
Out[4]: age          int64
sex          int64
cp          int64
trestbps     int64
chol         int64
fbs         int64
restecg      int64
thalach      int64
exang        int64
oldpeak     float64
slope        int64
ca          int64
thal        int64
target      int64
dtype: object
```

## Проверка на наличие пропусков в данных

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

```
Out[5]: age          0
sex          0
cp          0
trestbps     0
chol         0
fbs         0
restecg      0
thalach      0
exang        0
oldpeak     0
slope        0
ca          0
thal        0
target      0
dtype: int64
```

**Пропусков в данных не обнаружено, следовательно производить удаление столбцов и строк не требуется ¶**

```
B [6]: # Первые 5 строк датасета
data.head()
```

```
Out[6]:
```

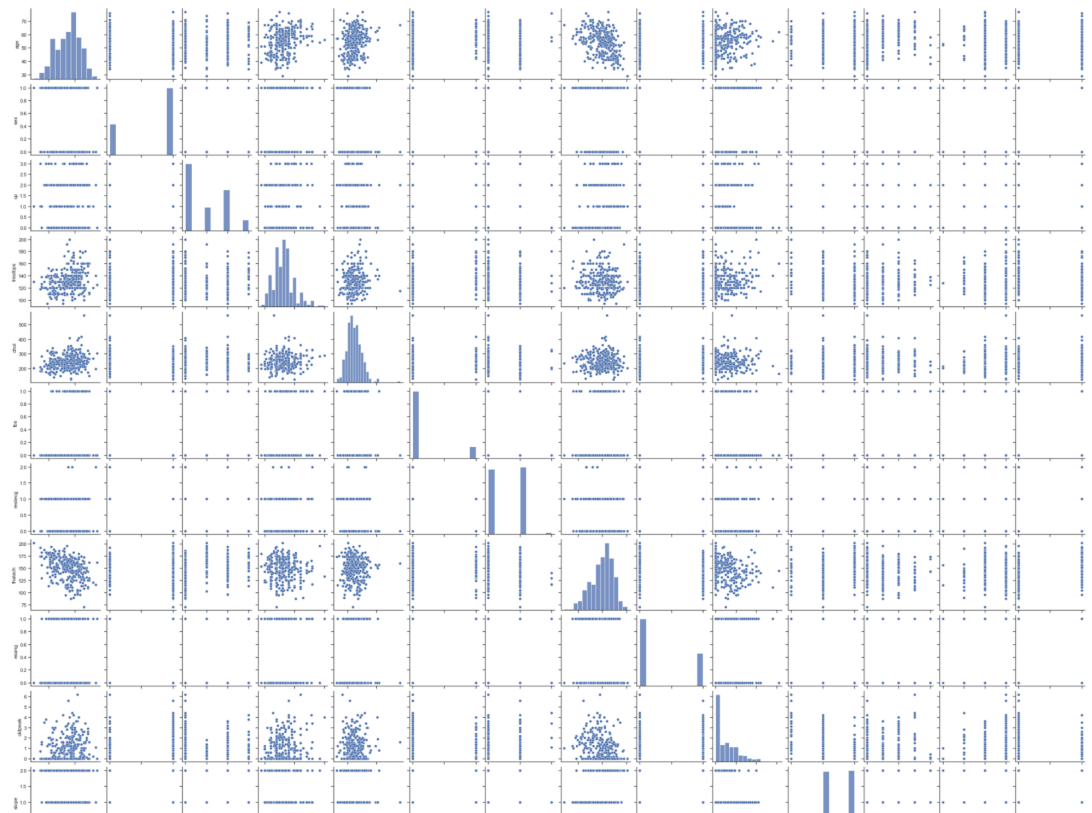
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

## "Парные диаграммы"

Построим диаграммы для данного датасета

```
B [7]: sns.pairplot(data)
```

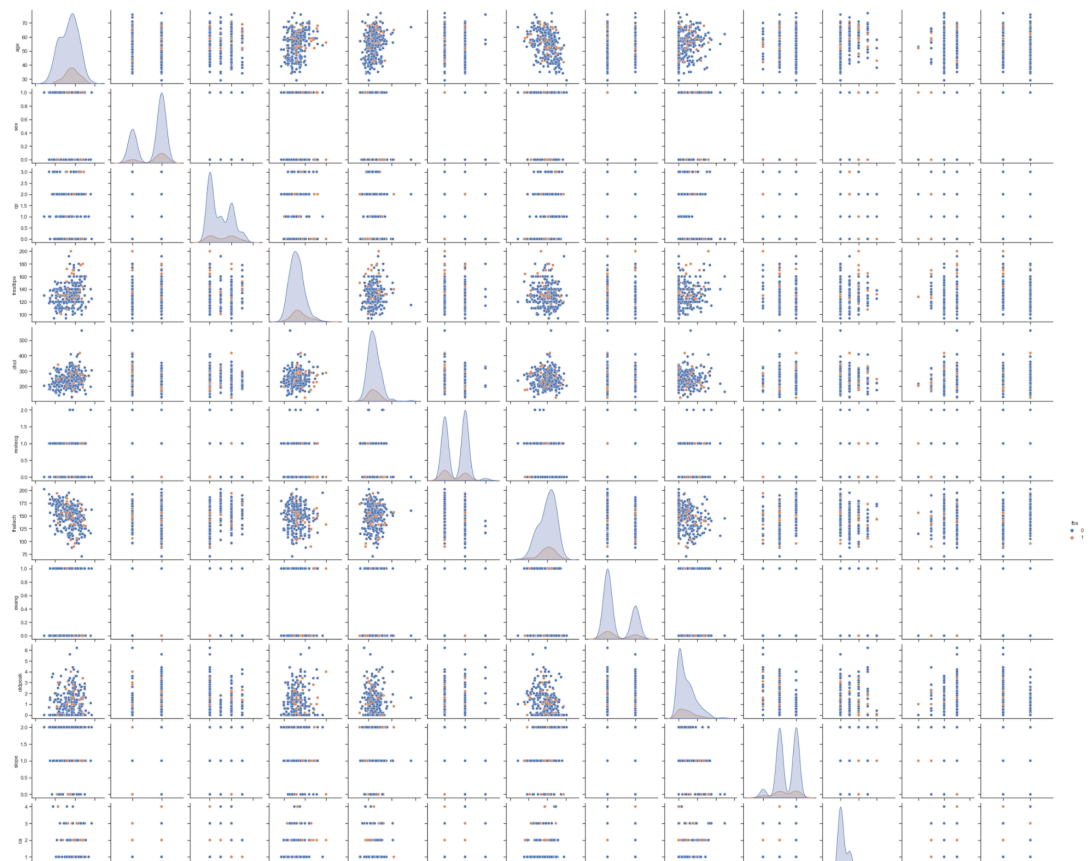
```
Out[7]: <seaborn.axisgrid.PairGrid at 0x7fc9d77dec10>
```



С помощью параметра "hue" построим диаграммы с группировкой по значениям признака illness

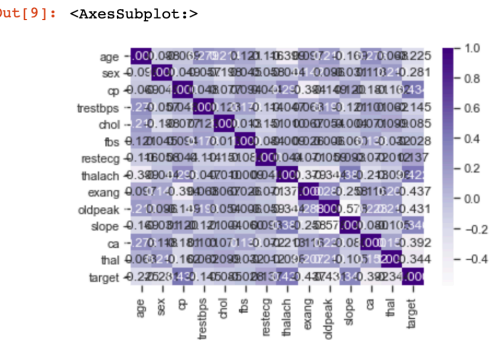
```
B [8]: sns.pairplot(data, hue="fbs")
```

```
Out[8]: <seaborn.axisgrid.PairGrid at 0x7fc9d2662cd0>
```



Корреляционный анализ

```
B [9]: sns.heatmap(data.corr(), cmap='Purples', annot=True, fmt='.3f')
```



```
B [10]: data.corr()
```

Out[10]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522	0.096801	0.210013	-0.168814	0.276326	0.068001	-0.225439
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020	0.141664	0.096093	-0.030711	0.118261	0.210041	-0.280937
cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.295762	-0.394280	-0.149230	0.119717	-0.181053	-0.161736	0.433798
trestbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.046698	0.067616	0.193216	-0.121475	0.101389	0.062210	-0.144931
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.009940	0.067023	0.053952	-0.004038	0.070511	0.098803	-0.085239
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.008567	0.025665	0.005747	-0.059894	0.137979	-0.032019	-0.028046
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.044123	-0.070733	-0.058770	0.093045	-0.072042	-0.011981	0.137230
thalach	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.000000	-0.378812	-0.344187	0.386784	-0.213177	-0.096439	0.421741
exang	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.378812	1.000000	0.288223	-0.257748	0.115739	0.206754	-0.436757
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.344187	0.288223	1.000000	-0.577537	0.222682	0.210244	-0.430696
slope	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.386784	-0.257748	-0.577537	1.000000	-0.080155	-0.104764	0.345877
ca	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.213177	0.115739	0.222682	-0.080155	1.000000	0.151832	-0.391724
thal	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.096439	0.206754	0.210244	-0.104764	0.151832	1.000000	-0.344029
target	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.421741	-0.436757	-0.430696	0.345877	-0.391724	-0.344029	1.000000