РК1 Вариант 5 Гасанов А.Ш. ИУ5-62Б

Задача №1.

Для заданного набора данных проведите корреляционный анализ. В случае наличия пропусков в данных удалите строки или колонки, содержащие пропуски. Сделайте выводы о возможности построения моделей машинного обучения и о возможном вкладе признаков в модель.

Колонки:

- age
- sex
- cp
- trestbps
- chol
- fbs
- restecg
- thalach
- exang
- oldpeak
- slope
- ca
- thal
- target

Resting blood pressure (trestbps) является целевым признаком.

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

In [2]:
```
data = pd.read_csv('heart.csv', sep=",")
```

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

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

In [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
```

In [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
```

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

Out[6]:

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

In [7]:
```
# Основные статистические характеристки набора данных
data.describe()
```

Out[7]:

	age	sex	cp	trestbps	chol	fbs	restecg
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000

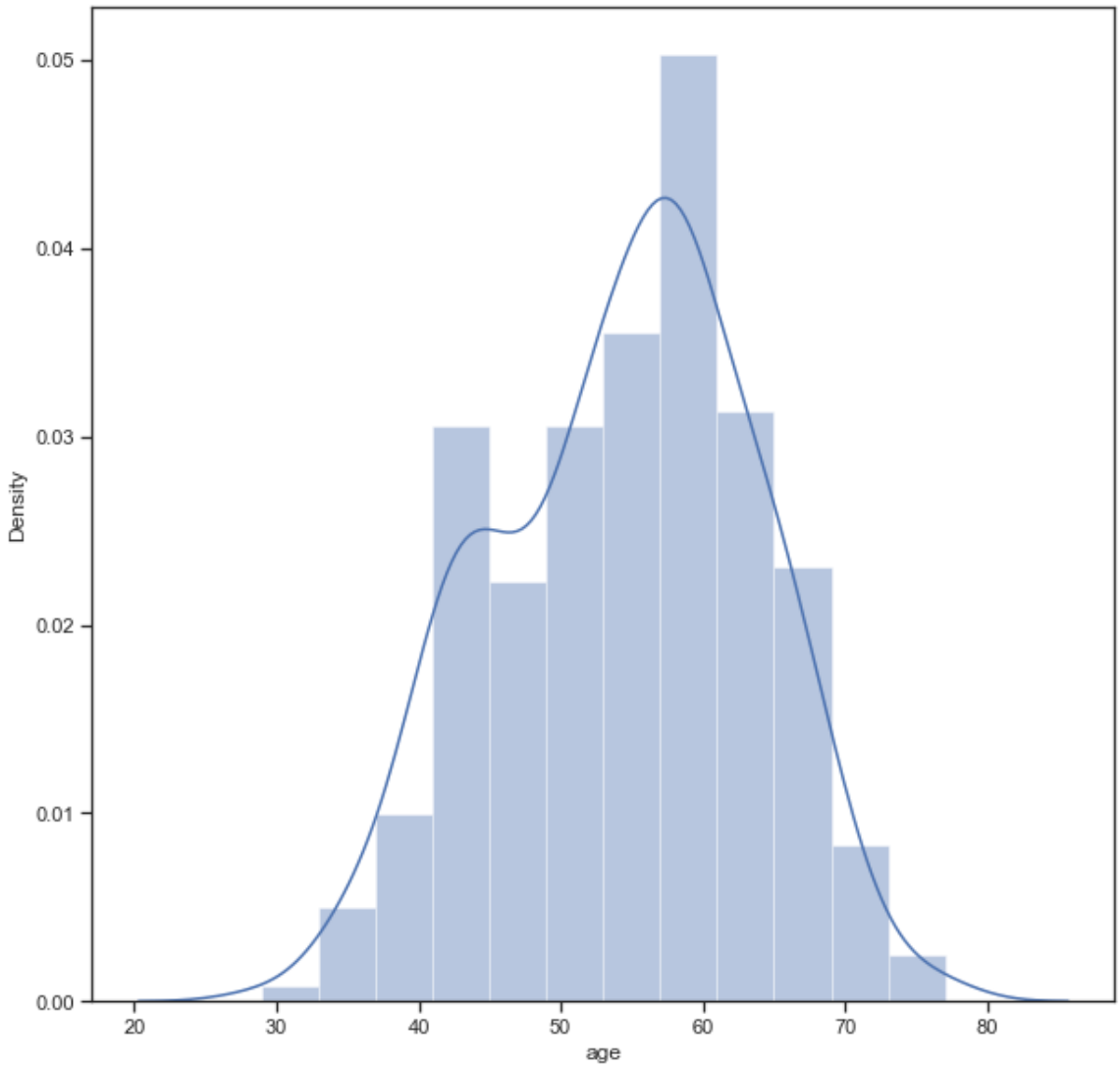
Гистограмма

Построим гистрограмму, которая позволит оценить плотность вероятности распределения данных.

In [14]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['age'])
```

```
c:\users\user\appdata\local\programs\python\python39\lib\site-packages\se
aborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated fu
nction and will be removed in a future version. Please adapt your code to
use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[14]: <AxesSubplot:xlabel='age', ylabel='Density'>



Информация о корреляции признаков

In [9]:
```
data.corr()
```

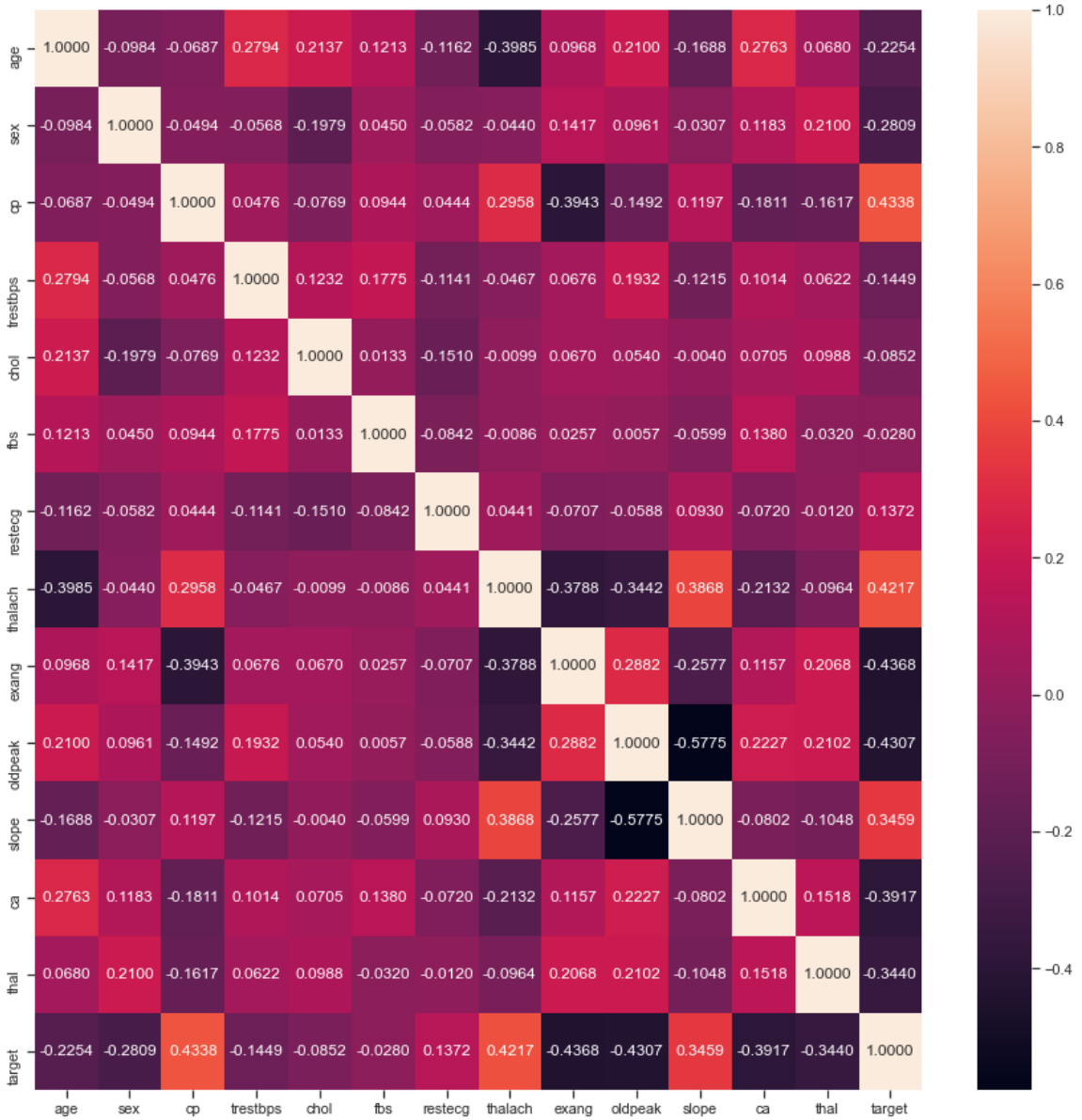
Out[9]:

	age	sex	cp	trestbps	chol	fbs	restecg	thala
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.3985
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.0440

cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.2957
trestbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.0466
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.0099
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.0085
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.0441
thalach	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.0000
exang	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.3788
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.3441
slope	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.3867
ca	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.2131
thal	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.0964
target	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.4217

In [10]:
```
fig, ax = plt.subplots(figsize=(15,15))
sns.heatmap(data.corr(), annot=True, fmt='.4f')
```

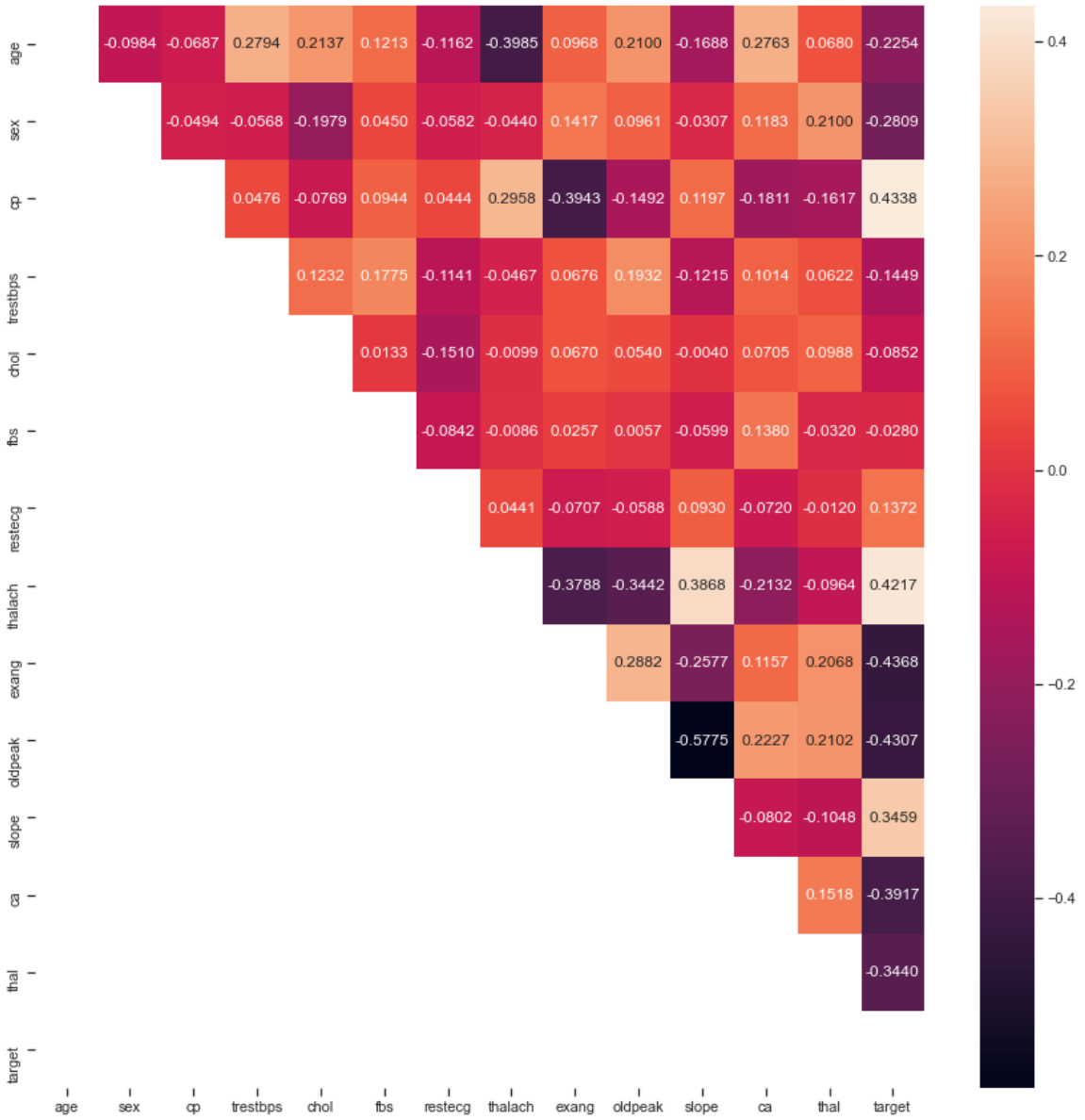
Out[10]: <AxesSubplot:>



In [12]:

```
# Вывод значений в ячейках
fig, ax = plt.subplots(figsize=(15,15))
mask = np.zeros_like(data.corr(), dtype=bool)
mask[np.tril_indices_from(mask)] = True
sns.heatmap(data.corr(), mask=mask, annot=True, fmt='.4f')
```

Out[12]: <AxesSubplot:>



Вывод

Наиболее коррелируемым признаком является ST depression induced by exercise relative to rest (oldpeak, 0.1932).