Московский государственный технический
университет им. Н.Э. Баумана

Факультет "Информатика и системы управления"
Кафедра ИУ5 "Системы обработки информации и управления"

Технологии машинного обучения

Отчет по лабораторной работе №1

Выполнил:
Студент группы ИУ5-65Б
Козинов Олег

Москва, 2021

Задание

Создать ноутбук, который содержит следующие разделы:

1. Текстовое описание выбранного Вами набора данных.
2. Основные характеристики датасета.
3. Визуальное исследование датасета.
4. Информация о корреляции признаков.

Выполнение лабораторной работы:

В лабораторной работе используется датасет Diabets.

1. Текстовое описание данных

   Данные о пациентах:

1. age- Возраст
2. sex- Пол
3. bmi- Индекс Массы Тела
4. bp- Кровяное давление
5. s1- Белые кровяные тела
6. s2- Липопротеины низкой плотности
7. s3- Липопротеины высокой плотности
8. s4- Тиреотропный гормон
9. s5- Ламотригин
10. s6- Уровень сахара в крови

Загрузка библиотек и датасета

In [2]:
```
import numpy as np
import pandas as pd
from sklearn.datasets import *
```

In [3]:
```
diabet = load_diabetes()
```

In [4]:
```
type(diabet)
```

Out[4]:
```
sklearn.utils.Bunch
```

Основные характеристики набора данных

In [5]:
```
for x in diabet:
    print(x)
```

```
data
target
frame
DESCR
feature_names
data_filename
target_filename
```

In [6]:
```
diabet['feature_names']
```

Out[6]:
```
['age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6']
```

In [7]:
```
diabet['data']
```

Out[7]:
```
array([[ 0.03807591,  0.05068012,  0.06169621, ..., -0.00259226,
         0.01990842, -0.01764613],
       [-0.00188202, -0.04464164, -0.05147406, ..., -0.03949338,
        -0.06832974, -0.09220405],
       [ 0.08529891,  0.05068012,  0.04445121, ..., -0.00259226,
         0.00286377, -0.02593034],
       ...,
       [ 0.04170844,  0.05068012, -0.01590626, ..., -0.01107952,
        -0.04687948,  0.01549073],
       [-0.04547248, -0.04464164,  0.03906215, ...,  0.02655962,
         0.04452837, -0.02593034],
       [-0.04547248, -0.04464164, -0.0730303 , ..., -0.03949338,
        -0.00421986,  0.00306441]])
```

In [8]:
```
diabet['data'].shape
```

Out[8]:
```
(442, 10)
```

In [9]:
```
data1 = pd.DataFrame(data= np.c_[diabet['data'], diabet['target']],
                     columns= diabet['feature_names'] + ['target'])
```

In [10]:
```
data1
```

Out[10]:

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019908	-0.017646	151.0
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068330	-0.092204	75.0
2	0.085299	0.050680	0.044451	-0.005671	-0.045599	-0.034194	-0.032356	-0.002592	0.002864	-0.025930	141.0
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022692	-0.009362	206.0
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031991	-0.046641	135.0
...	...	...	...	...	...	...	...	...	...	...	...
437	0.041708	0.050680	0.019662	0.059744	-0.005697	-0.002566	-0.028674	-0.002592	0.031193	0.007207	178.0
438	-0.005515	0.050680	-0.015906	-0.067642	0.049341	0.079165	-0.028674	0.034309	-0.018118	0.044485	104.0
439	0.041708	0.050680	-0.015906	0.017282	-0.037344	-0.013840	-0.024993	-0.011080	-0.046879	0.015491	132.0
440	-0.045472	-0.044642	0.039062	0.001215	0.016318	0.015283	-0.028674	0.026560	0.044528	-0.025930	220.0
441	-0.045472	-0.044642	-0.073030	-0.081414	0.083740	0.027809	0.173816	-0.039493	-0.004220	0.003064	57.0

442 rows × 11 columns

In [11]:
```
data1.head()
```

Out[11]:

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019908	-0.017646	151.0
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068330	-0.092204	75.0
2	0.085299	0.050680	0.044451	-0.005671	-0.045599	-0.034194	-0.032356	-0.002592	0.002864	-0.025930	141.0
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022692	-0.009362	206.0
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031991	-0.046641	135.0

In [12]:
```
data1.shape
```

Out[12]:
```
(442, 11)
```

In [13]:
```
total_count = data1.shape[0]
print('Всего строк: {}'.format(total_count))
```

```
Всего строк: 442
```

In [14]:
```
data1.columns
```

Out[14]:
```
Index(['age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6',
       'target'],
      dtype='object')
```

In [15]:
```
data1.dtypes
```

Out[15]:
```
age       float64
sex       float64
bmi       float64
bp        float64
s1        float64
s2        float64
s3        float64
s4        float64
s5        float64
s6        float64
target    float64
dtype: object
```

In [16]:
```
for col in data1.columns:
    temp_null_count = data1[data1[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
age - 0
sex - 0
bmi - 0
bp - 0
s1 - 0
s2 - 0
s3 - 0
s4 - 0
s5 - 0
s6 - 0
target - 0
```

In [17]:
```
data1.describe()
```

Out[17]:

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	targ
count	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	442.0000
mean	-3.634285e-16	1.308343e-16	-8.045349e-16	1.281655e-16	-8.835316e-17	1.327024e-16	-4.574646e-16	3.777301e-16	-3.830854e-16	-3.412882e-16	152.1334
std	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	77.0930
min	-1.072256e-01	-4.464164e-02	-9.027530e-02	-1.123996e-01	-1.267807e-01	-1.156131e-01	-1.023071e-01	-7.639450e-02	-1.260974e-01	-1.377672e-01	25.0000
25%	-3.729927e-02	-4.464164e-02	-3.422907e-02	-3.665645e-02	-3.424784e-02	-3.035840e-02	-3.511716e-02	-3.949338e-02	-3.324879e-02	-3.317903e-02	87.0000
50%	5.383060e-03	-4.464164e-02	-7.283766e-03	-5.670611e-03	-4.320866e-03	-3.819065e-03	-6.584468e-03	-2.592262e-03	-1.947634e-03	-1.077698e-03	140.5000
75%	3.807591e-02	5.068012e-02	3.124802e-02	3.564384e-02	2.835801e-02	2.984439e-02	2.931150e-02	3.430886e-02	3.243323e-02	2.791705e-02	211.5000
max	1.107267e-01	5.068012e-02	1.705552e-01	1.320442e-01	1.539137e-01	1.987880e-01	1.811791e-01	1.852344e-01	1.335990e-01	1.356118e-01	346.0000

In [18]:
```
data1['age'].unique()
```

Out[18]:
```
array([ 0.03807591, -0.00188202,  0.08529891, -0.08906294,  0.00538306,
       -0.09269548, -0.04547248,  0.06350368,  0.04170844, -0.07090025,
       -0.09632802,  0.02717829,  0.01628068,  0.04534098, -0.05273755,
       -0.00551455,  0.07076875, -0.0382074 , -0.02730979, -0.04910502,
       -0.0854304 , -0.06363517, -0.06726771, -0.10722563, -0.02367725,
        0.05260606,  0.06713621, -0.06000263,  0.03444337,  0.03081083,
        0.04897352,  0.01264814, -0.00914709, -0.09996055,  0.01991321,
       -0.05637009, -0.07816532, -0.04183994,  0.05987114, -0.03457486,
       -0.03094232, -0.10359309, -0.01641217,  0.00175052, -0.02004471,
        0.0562386 ,  0.02354575,  0.0090156 , -0.07453279, -0.01277963,
       -0.08179786,  0.08166637,  0.11072668,  0.09256398,  0.07440129,
        0.07803383,  0.09619652,  0.08893144])
```

Визуальное исследование датасета

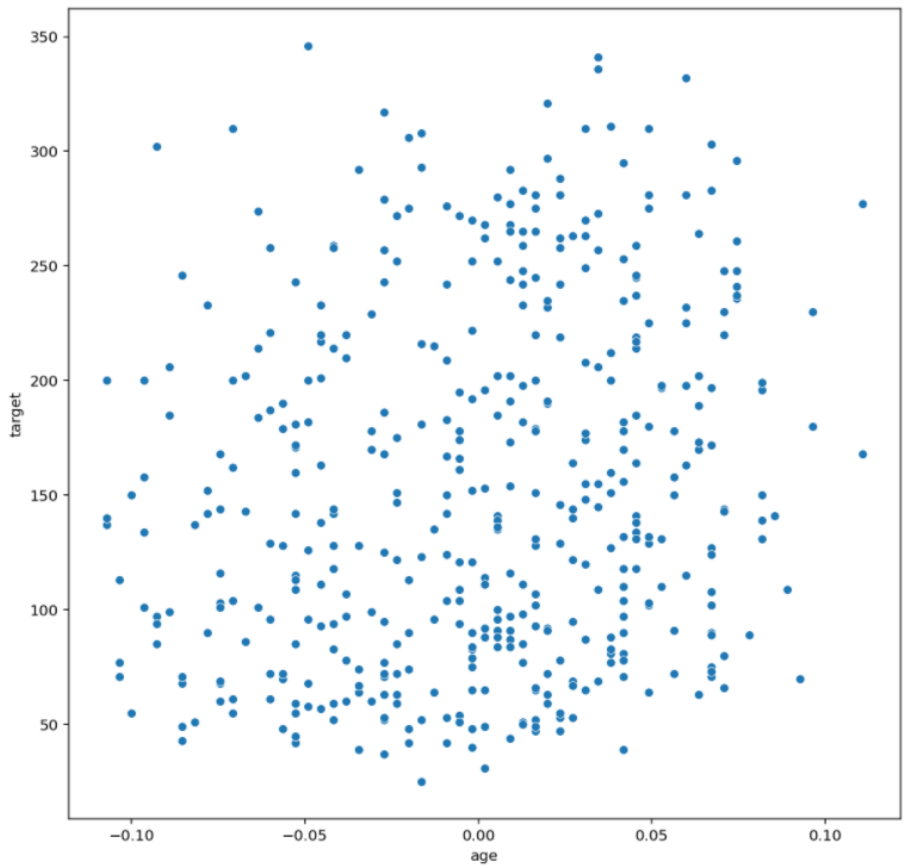
```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set(style="ticks")
```

Диаграмма рассеивания

In [19]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='age', y='target', data=data1)
```

Out[19]:
```
<AxesSubplot:xlabel='age', ylabel='target'>
```

Out[19]:



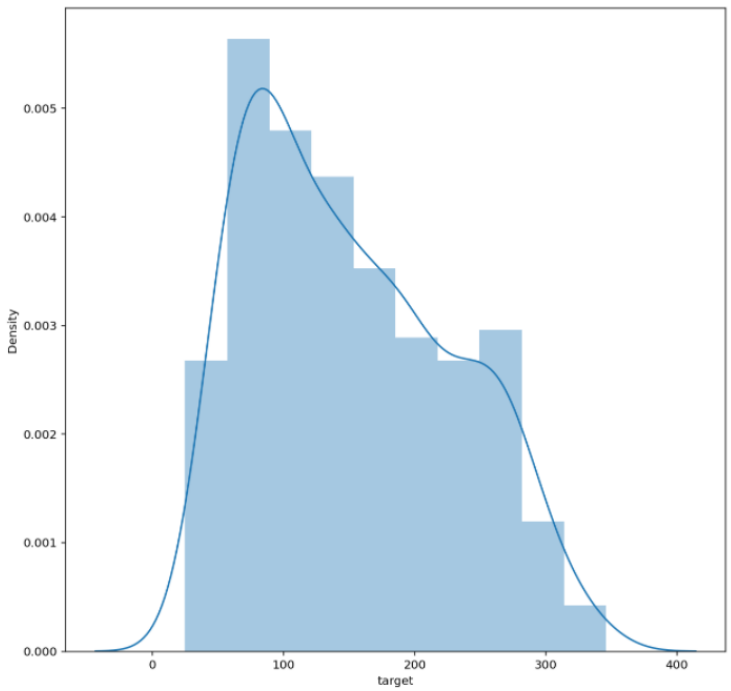
Гистограмма

In [21]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data1['target'])
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a deprecated function and wi
ll be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibi
lity) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[21]: <AxesSubplot:xlabel='target', ylabel='Density'>

Out[21]:

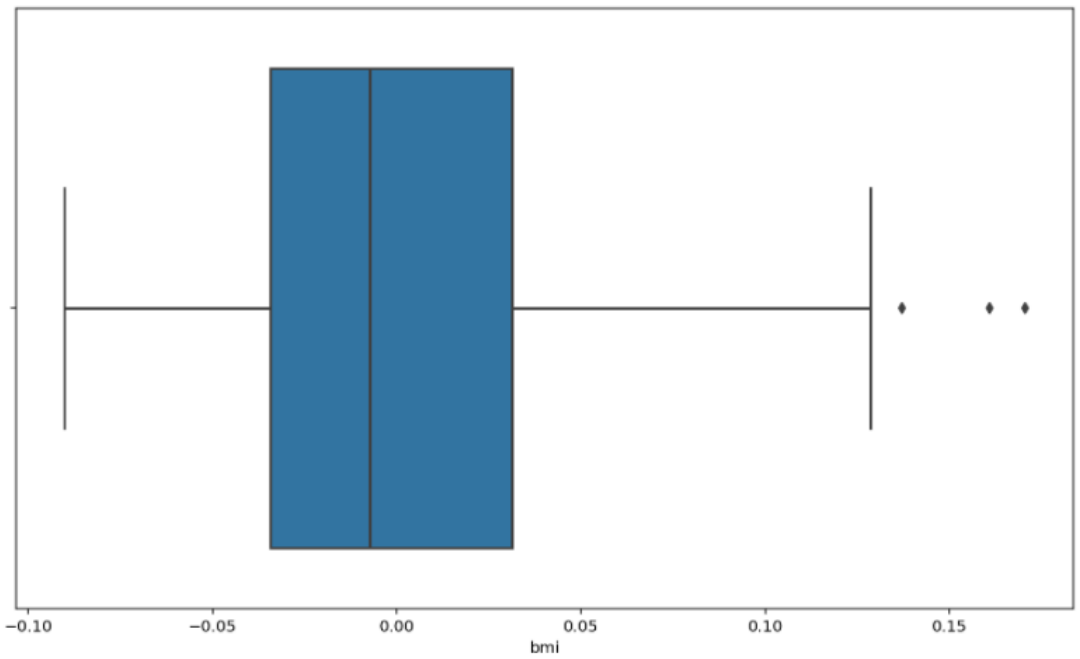


Ящик с усами

In [24]:
```
sns.boxplot(x=data1['bmi'])
```

Out[24]: <AxesSubplot:xlabel='bmi'>

Out[24]:



Информация о корреляции признаков

In [25]:
```
data1.corr()
```

Out[25]:

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
age	1.000000	0.173737	0.185085	0.335427	0.260061	0.219243	-0.075181	0.203841	0.270777	0.301731	0.187889
sex	0.173737	1.000000	0.088161	0.241013	0.035277	0.142637	-0.379090	0.332115	0.149918	0.208133	0.043062
bmi	0.185085	0.088161	1.000000	0.395415	0.249777	0.261170	-0.366811	0.413807	0.446159	0.388680	0.586450
bp	0.335427	0.241013	0.395415	1.000000	0.242470	0.185558	-0.178761	0.257653	0.393478	0.390429	0.441484
s1	0.260061	0.035277	0.249777	0.242470	1.000000	0.896663	0.051519	0.542207	0.515501	0.325717	0.212022
s2	0.219243	0.142637	0.261170	0.185558	0.896663	1.000000	-0.196455	0.659817	0.318353	0.290600	0.174054
s3	-0.075181	-0.379090	-0.366811	-0.178761	0.051519	-0.196455	1.000000	-0.738493	-0.398577	-0.273697	-0.394789
s4	0.203841	0.332115	0.413807	0.257653	0.542207	0.659817	-0.738493	1.000000	0.617857	0.417212	0.430453
s5	0.270777	0.149918	0.446159	0.393478	0.515501	0.318353	-0.398577	0.617857	1.000000	0.464670	0.565883
s6	0.301731	0.208133	0.388680	0.390429	0.325717	0.290600	-0.273697	0.417212	0.464670	1.000000	0.382483
target	0.187889	0.043062	0.586450	0.441484	0.212022	0.174054	-0.394789	0.430453	0.565883	0.382483	1.000000

Тепловая карта

In [26]:
```
sns.heatmap(data1.corr(), annot=True)
```

Out[26]:
```
<AxesSubplot:>
```

Out[26]: