



**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 «Системы обработки информации и управления»**

**Отчет по лабораторной работе № 1
по дисциплине «Технология машинного обучения»**

Выполнил:
студент группы ИУ5-63Б Кузнецов В.А.
подпись, дата

Проверил:
Гапанюк Ю.Е.
подпись, дата

2024 г.

Задание:

1. Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов [здесь](#).
 - a. Для первой лабораторной работы рекомендуется использовать датасет без пропусков в данных, например из Scikit-learn.
 - b. Пример преобразования датасетов Scikit-learn в Pandas Dataframe можно посмотреть [здесь](#).
 - c. Для лабораторных работ не рекомендуется выбирать датасеты большого размера.
2. Создать ноутбук, который содержит следующие разделы:
 - a. Текстовое описание выбранного Вами набора данных.
 - b. Основные характеристики датасета.
 - c. Визуальное исследование датасета.
 - d. Информация о корреляции признаков.
3. Сформировать отчет и разместить его в своей репозитории на github.

Текст программы:

✓ Текстовое описание

Датасет с данными о диабете, включённый в библиотеку scikit-learn, является широко используемым набором данных для задач регрессии.

Он содержит информацию о человеке: возраст, пол, индекс массы тела, среднее кровяное давление и шесть измерений сыворотки крови.

Целевая переменная является количественным показателем прогрессирования заболевания через год после исходного уровня, который, по сути, является показателем тяжести или прогрессирования заболевания.

✓ Основные характеристики датасета

age: Возраст в годах

sex: Пол

bmi: индекс массы тела, показатель жировых отложений, основанный на росте и весе

bp: Среднее кровяное давление

S1: Общий холестерин сыворотки крови

S2: Холестерин липопротеидов низкой плотности (ЛПНП)

S3: Холестерин липопротеидов высокой плотности (ЛПВП)

S4: Соотношение общего холестерина и холестерина ЛПВП

S5: Логарифм уровня триглицеридов в сыворотке крови

S6: Уровень сахара в крови

target: Целевая переменная - Количественный показатель прогрессирования заболевания

✓ Импорт библиотек

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_diabetes
import seaborn as sns
import matplotlib.pyplot as plt
```

✓ Загрузка данных


```
diabetes = load_diabetes()
data = pd.DataFrame(data= np.c_[diabetes['data'], diabetes['target']],
                    columns= diabetes['feature_names'] + ['target'])
```

✓ Основные характеристики датасета

```
print("Всего строк:", data.shape[0])
print("Всего столбцов:", data.shape[1])
```


↗ Всего строк: 442
Всего столбцов: 11

```
data.head()
```




	age	sex	bmi	bp	s1	s2	s3	s4
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592

```
# Список колонок с типами данных
data.dtypes
```




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

```
# Пропуски по колонкам
for col in data.columns:
    temp_null_count = data[data[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
```

```
# Основные статистические характеристики набора данных
data.describe()
```




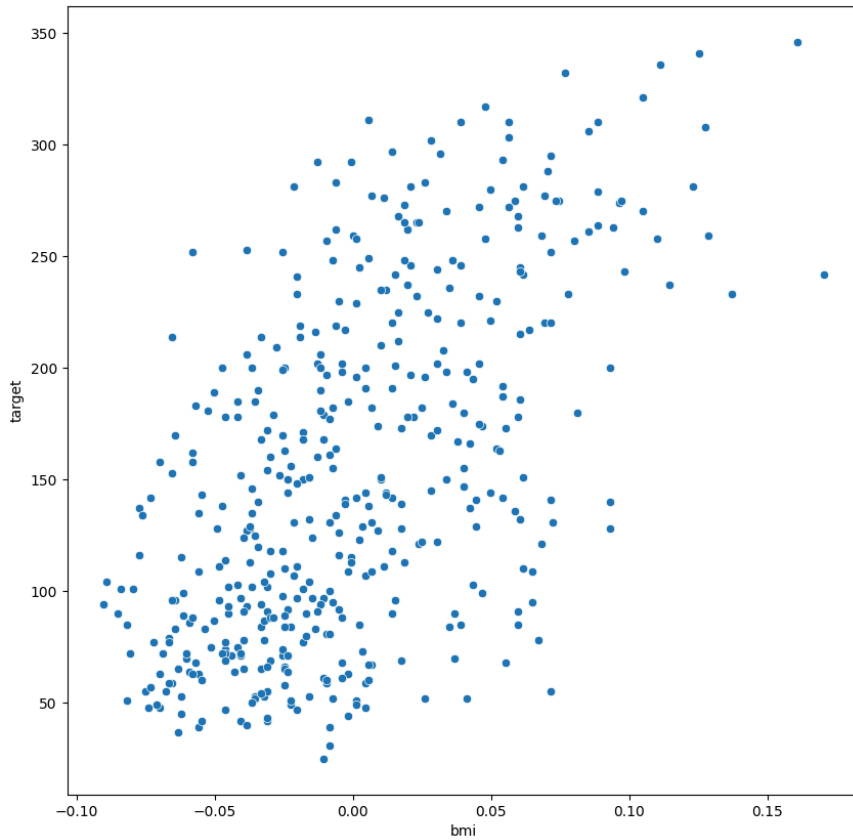
	age	sex	bmi	bp	s1	s2	s3	s4
count	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02
mean	-2.511817e-19	1.230790e-17	-2.245564e-16	-4.797570e-17	-1.381499e-17	3.918434e-17	-1.156131e-17	-1.156131e-17
std	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02
min	-1.072256e-01	-4.464164e-02	-9.027530e-02	-1.123988e-01	-1.267807e-01	-1.156131e-01	-1.156131e-01	-1.156131e-01
25%	-3.729927e-02	-4.464164e-02	-3.422907e-02	-3.665608e-02	-3.424784e-02	-3.035840e-02	-3.035840e-02	-3.035840e-02
50%	5.383060e-03	-4.464164e-02	-7.283766e-02	-5.670422e-02	-4.320866e-02	-3.819060e-02	-3.819060e-02	-3.819060e-02

Можно заметить, что данные уже масштабированы.


✓ Визуальное исследование

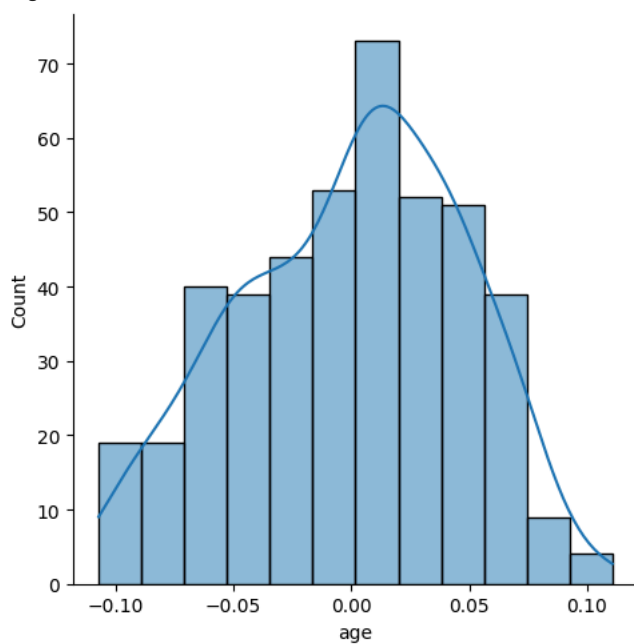
```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='bmi', y='target', data=data)
```

 <Axes: xlabel='bmi', ylabel='target'>




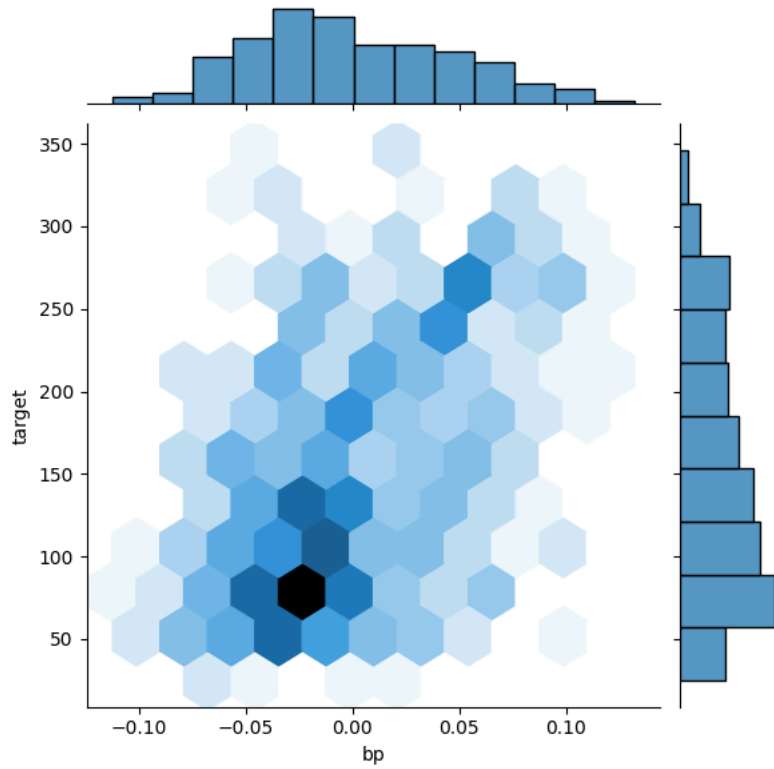
```
plt.figure(figsize=(10, 10))
sns.displot(data['age'], kde=True)
plt.show()
```

 <Figure size 1000x1000 with 0 Axes>



```
sns.jointplot(x='bp', y='target', data=data, kind="hex")
```

 <seaborn.axisgrid.JointGrid at 0x7c8c9e165060>



```
sns.pairplot(data)
```



<seaborn.axisgrid.PairGrid at 0x7c8c9dfb9930>

