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, искусственный интеллект и системы управления

КАФЕДРА Системы обработки информации и управления

Отчет по рубежному контролю №1 по курсу «Методы машинного обучения»

Рубежный контроль №1

«Методы обработки данных»

Выполнил:
Тураев Г.В.
ИУ5-25М

Проверил:

Гапанюк Ю.Е

Москва, 2023

Условие.

Номер по списку группы (вариант): 14.

Вариант задачи №1: 14. Вариант задачи №2: 34.

Условие задачи №1:

Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием функции "квадратный корень".

Условие задачи №2:

Для набора данных проведите процедуру отбора признаков (feature selection). Используйте метод вложений (embedded method). Используйте подход на основе линейной или логистической регрессии (в зависимости от того, на решение какой задачи ориентирован выбранный Вами набор данных - задачи регрессии или задачи классификации).

Дополнительное задание (по группам):

Для студентов группы ИУ5-25М – для произвольной колонки данных построить парные диаграммы (pairplot).

Выполнение.

Для выполнения рубежного контроля используется датасет, связанный с «Покемонами».

Исследуемый набор данных состоит из 410 строк и 11 колонок. Описание столбцов представлено в таблице 1. Целевыми столбцами в наших исследованиях являются attack, defense, sp. atc.

Таблица – 1. Описание колонок набора данных

Название колонки	Описание
ID	Уникальный номер покемона
Name	Имя покемона
Type 1	Тип, который определяет слабость или устойчивость к атакам
Type 2	Тип, который определяет слабость или устойчивость к атакам
Total	Рейтинг (показывает, насколько силен покемон)
HP	Здоровье (показывает, сколько урона может выдержать покемон)
Attack	Атака (базовый модификатор для обычных атак, например, царапина или удар)
Defense	Защита (базовая устойчивость к урону от обычных атак)
Special attack	Специальная атака (базовый модификатор для специальных атак, например, огненный взрыв или пузырьковый луч)
Special defense	Специальная защита (базовая устойчивость к урону от специальных атак)
Speed	Скорость

Задача №1 (задача №14).

Импортируем библиотеки.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sbn
from sklearn. linear_model import LogisticRegression
from sklearn.feature_selection import SelectFromModel
from sklearn.svm import LinearSVC
```

```
data = pd.read_excel('Pokemon.xls')
```

```
data.head()
```

	ID	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2	3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	4	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	5	Charizard	Fire	Flying	534	78	84	78	109	85	100	1	False

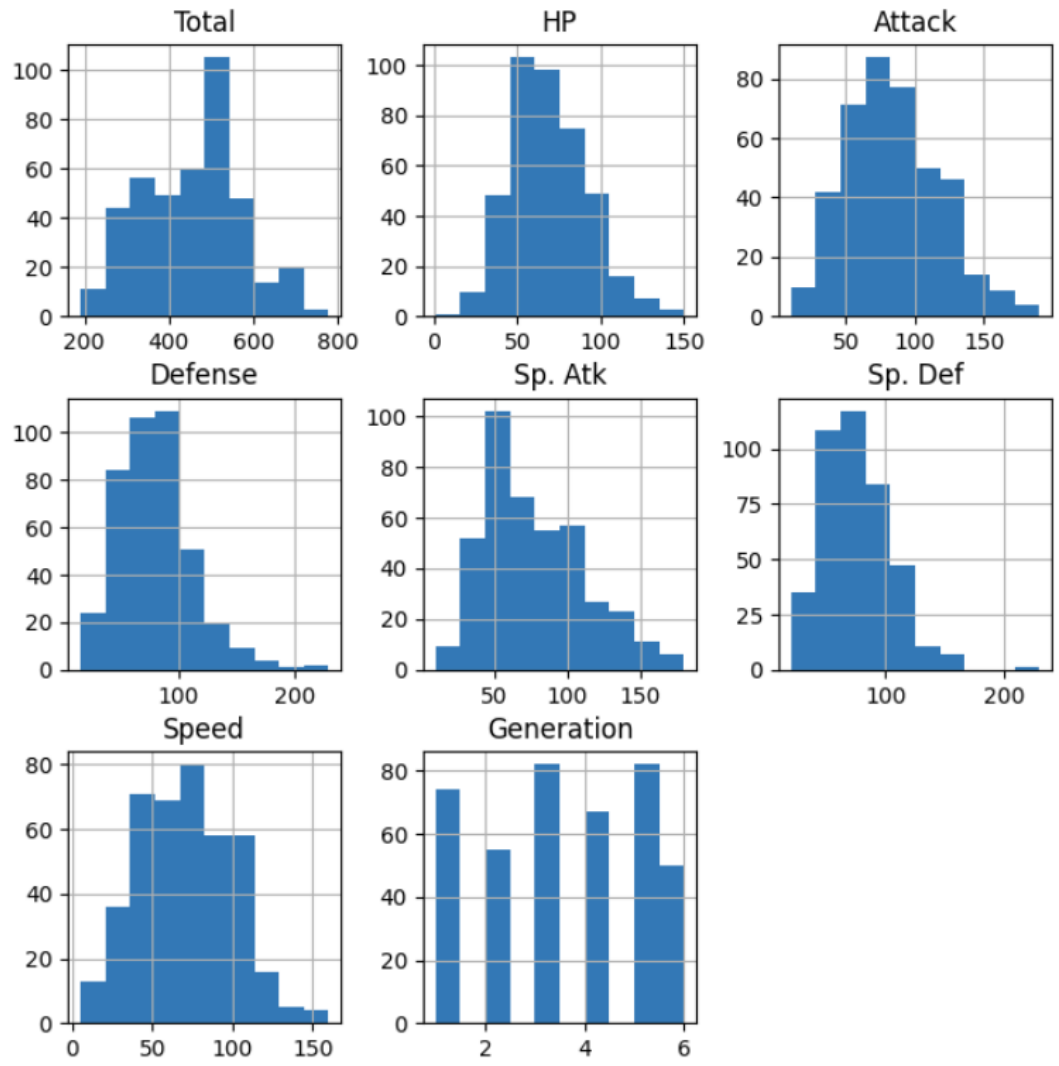
```
data = data.drop('ID', 1)
data.head()
```

```
<ipython-input-28-c6a65ae19bdd>:1: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the
  data = data.drop('ID', 1)
```

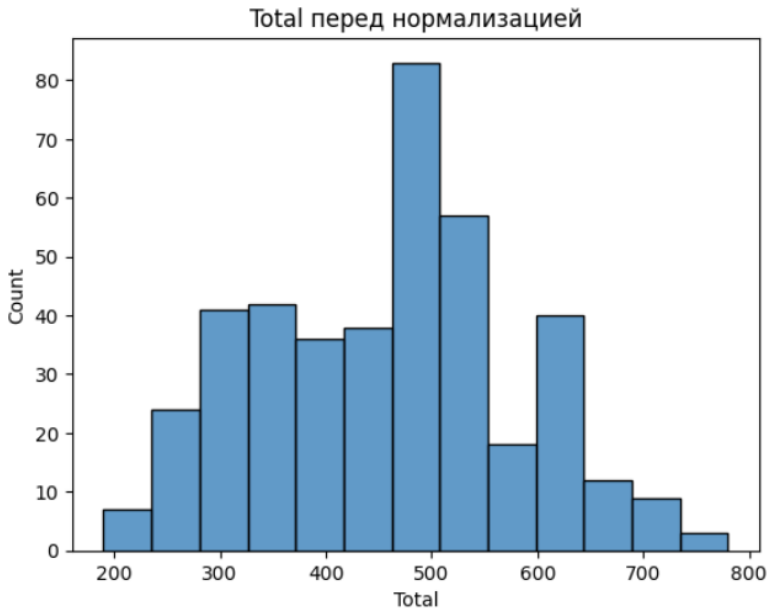
	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
1	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
2	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	Charizard	Fire	Flying	534	78	84	78	109	85	100	1	False

Выведем распределения признаков.

```
data.hist(figsize=(8,8))
plt.show()
```



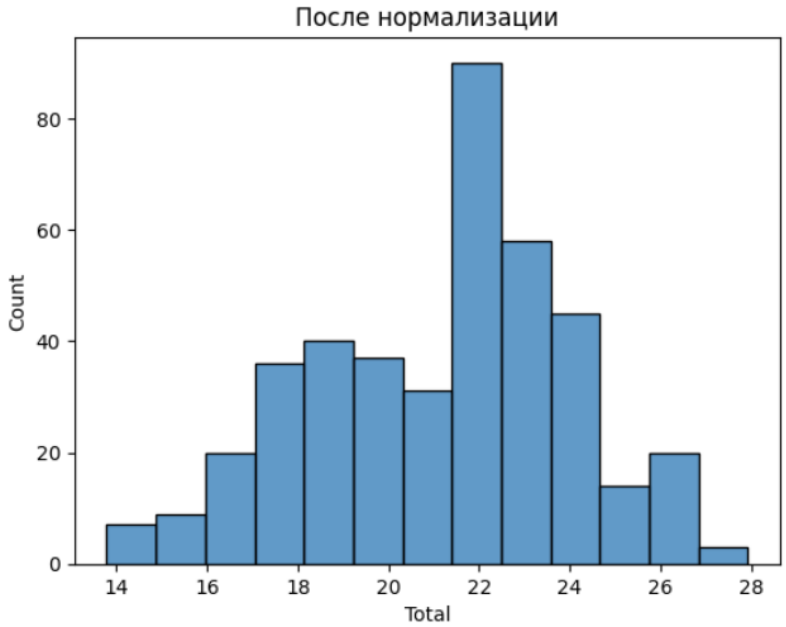
```
normalization = 'Total'
sbn.histplot(data[normalization])
plt.title(f'{normalization} перед нормализацией')
plt.show()
```



Как видим, оно немного отличается от нормального распределения.

Произведем нормализацию с помощью функции «квадратный корень».

```
data[normalization] = data[normalization]**(1/2)
sbn.histplot(data[normalization])
plt.title(f'После нормализации')
plt.show()
```



Как итог, удалось с помощью функции «квадратный корень» успешно нормализовать признак «Total».

Задача №2 (задача №34).

```
data = data.drop('Name', 1)
data.head()
```

```
<ipython-input-39-868879867e4c>:1: FutureWarning: In a future version of pandas all arguments of DataFrame
  data = data.drop('Name', 1)
```

	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	Grass	Poison	17.832555	45	49	49	65	65	45	1	False
1	Grass	Poison	20.124612	60	62	63	80	80	60	1	False
2	Grass	Poison	22.912878	80	82	83	100	100	80	1	False
3	Grass	Poison	25.000000	80	100	123	122	120	80	1	False
4	Fire	Flying	23.108440	78	84	78	109	85	100	1	False

```
data = data.drop('Type 1', 1)
data.head()
```

```
<ipython-input-42-a349c36924c2>:1: FutureWarning: In a future version of pandas all arguments of DataFrame
  data = data.drop('Type 1', 1)
```

	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	Poison	17.832555	45	49	49	65	65	45	1	False
1	Poison	20.124612	60	62	63	80	80	60	1	False
2	Poison	22.912878	80	82	83	100	100	80	1	False
3	Poison	25.000000	80	100	123	122	120	80	1	False
4	Flying	23.108440	78	84	78	109	85	100	1	False

```
data = data.drop('Type 2', 1)
data.head()
```

```
<ipython-input-45-da43cef176cd>:1: FutureWarning: In a future version of pandas all argume
  data = data.drop('Type 2', 1)
```

	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
0	17.832555	45	49	49	65	65	45	1	False
1	20.124612	60	62	63	80	80	60	1	False
2	22.912878	80	82	83	100	100	80	1	False
3	25.000000	80	100	123	122	120	80	1	False
4	23.108440	78	84	78	109	85	100	1	False

```
wine_X = data.drop('Speed', 1).values
wine_y= data['Speed'].values
wine_feature_names = list(data.drop('Speed', 1).keys())
wine_x_df = pd.DataFrame(data=wine_X, columns=wine_feature_names)
```

```
<ipython-input-46-fb69ff1b7f33>:1: FutureWarning: In a future version of pandas all argume
  wine_X = data.drop('Speed', 1).values
<ipython-input-46-fb69ff1b7f33>:3: FutureWarning: In a future version of pandas all argume
  wine_feature_names = list(data.drop('Speed', 1).keys())
```

Выведем коэффициенты регрессии, используя логистическую регрессию.

```
e_lr1 = LogisticRegression(C=1000, solver='liblinear', penalty='l1', max_iter=500, random_state=1)
e_lr1.fit(wine_X, wine_y)
e_lr1.coef_
```

```
       -5.40108919e-02,  1.57960919e+00],
      [ 1.50523435e+00, -4.03930878e-01,  2.28487183e-01,
       -1.07465316e+00, -3.38913856e-01, -5.18361810e-01,
        6.15781445e+00,  1.01943410e+02],
      [ 4.38292882e+00, -2.64315049e-01, -3.48414214e-02,
       -5.11890661e-02, -6.09169524e-02, -1.37180265e-01,
        1.05516140e+00, -1.19055137e+01],
      [ 2.03908545e+01, -5.04151087e-01, -1.20598572e+00,
       -3.64029225e-01, -4.76068017e-01, -7.20889785e-01,
        2.55548736e+00,  0.00000000e+00],
      [ 1.93115516e+00, -8.55701715e-02, -2.56121297e-02,
       -4.25846095e-02, -2.21314878e-02, -4.81008438e-02,
       -6.54920269e-01,  9.53654248e-01],

                          .............

       [ 1.18670431e+01, -4.04537155e-01, -4.13963968e-02,
        -5.81592014e-01, -9.25633996e-02, -1.99277819e-01,
        -1.12193862e+00, -9.61659605e+00],
       [ 1.06998236e+01, -8.73373976e-02, -2.27856319e-01,
        -1.11514911e-01, -2.52133021e-01, -1.46326493e-01,
        -9.06561832e+01, -5.35657515e+00],
       [ 7.89144665e+00, -1.26182284e-01, -2.18727676e-01,
        -3.65652737e-01, -2.41120084e-01, -4.26400134e-01,
         2.52273746e-01,  0.00000000e+00]])
```

Как итог, данные данного датасета являются «хорошими».

```
sel_e_lr1 = SelectFromModel(e_lr1)
sel_e_lr1.fit(wine_X, wine_y)
for feature, flag in zip(wine_feature_names, sel_e_lr1.get_support()):
  print(feature, ':', flag)
```

```
Total : True
HP : True
Attack : True
Defense : True
Sp. Atk : True
Sp. Def : True
Generation : True
Legendary : True
/usr/local/lib/python3.9/dist-packages/sklearn/svm/_base.py:1244: Convergence
  warnings.warn(
```

Выведем коэффициенты регрессии, используя линейный классификатор на основе SVM.

```
e_lr2 = LinearSVC(C=0.01, penalty="l1", max_iter=2000, dual=False)
e_lr2.fit(wine_X, wine_y)
# Коэффициенты регрессии
e_lr2.coef_
```

```
         0.00000000e+00,  0.00000000e+00],
       [ 0.00000000e+00, -8.16019842e-03,  0.00000000e+00,
        -8.68484659e-03,  0.00000000e+00, -4.16765820e-03,
         0.00000000e+00,  0.00000000e+00],
       [-1.19245661e-02, -1.02006992e-02,  8.85644711e-04,
        -9.90842573e-04,  0.00000000e+00, -5.08401863e-03,
         0.00000000e+00,  0.00000000e+00],
       [ 0.00000000e+00, -4.49995686e-03, -6.05798424e-03,
        -3.47942066e-03,  0.00000000e+00, -7.29214292e-03,
         0.00000000e+00,  0.00000000e+00],
       [-6.14931318e-02, -1.85305611e-03,  1.91810505e-03,
         0.00000000e+00,  3.20121093e-03,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00],

                          .............

       [ 0.00000000e+00, -9.75337471e-03,  3.28736862e-03,
        -1.35736703e-02, -1.13474267e-03,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00],
       [ 0.00000000e+00, -7.53002239e-03,  0.00000000e+00,
        -5.13763104e-03, -8.13391272e-03,  0.00000000e+00,
         0.00000000e+00,  0.00000000e+00],
       [ 0.00000000e+00, -4.15047464e-03,  0.00000000e+00,
        -1.05977321e-02, -5.69668626e-03, -2.11471887e-03,
         0.00000000e+00,  0.00000000e+00]])
```

Для дальнейшей работы оставим только те признаки, которые являются с флагом True.

```
sel_e_lr2 = SelectFromModel(e_lr2)
sel_e_lr2.fit(wine_X, wine_y)
for feature, flag in zip(wine_feature_names, sel_e_lr2.get_support()):
  print(feature, ':', flag)
```

```
Total : True
HP : True
Attack : True
Defense : True
Sp. Atk : True
Sp. Def : True
Generation : True
Legendary : False
```

Дополнительное задание (для группы ИУ5-25М).

Построим парные диаграммы.

```
sbn.pairplot(data, hue = 'Generation')
```

```
<seaborn.axisgrid.PairGrid at 0x7fb4de0b2340>
```