# Лабораторная работа №1.

## Исследование набора данных ирисы Фишера.

### Подключение библиотеки NumPy и загрузка данных

```
!wget
https://raw.githubusercontent.com/cnrde/AI_and_ML_1/main/dataset/set.d
ata

--2024-05-13 18:01:06--
https://raw.githubusercontent.com/cnrde/AI_and_ML_1/main/dataset/set.d
ata
Resolving raw.githubusercontent.com (raw.githubusercontent.com)...
185.199.110.133, 185.199.108.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|
185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3245 (3.2K) [text/plain]
Saving to: ‘set.data’

set.data            100%[===================>]   3.17K  --.-KB/s    in
0s

2024-05-13 18:01:07 (35.0 MB/s) - ‘set.data’ saved [3245/3245]
```

```
import numpy as np
data_path = "set.data"
data = np.genfromtxt(data_path, delimiter=",")
print(data)
```

```
[[7.2 3.6 6.1 2.5 nan]
 [6.3 3.3 6.  2.5 nan]
 [6.7 3.  5.2 2.3 nan]
 [6.5 3.  5.8 2.2 nan]
 [7.1 3.  5.9 2.1 nan]
 [6.6 3.  4.9 1.8 nan]
 [7.3 2.9 6.3 1.8 nan]
 [6.7 2.5 5.8 1.8 nan]
 [7.2 3.6 6.1 2.5 nan]
 [6.3 2.5 5.  1.9 nan]
 [7.2 3.2 6.  1.8 nan]
 [6.2 3.4 5.4 2.3 nan]
 [7.  3.2 6.  1.8 nan]
 [6.9 3.1 5.1 2.3 nan]
 [6.4 2.7 5.3 1.9 nan]
 [6.2 2.2 4.5 1.5 nan]
 [6.8 3.  5.5 2.1 nan]
 [6.  2.7 5.1 1.6 nan]
 [5.6 2.5 3.9 1.1 nan]
 [6.5 3.  5.5 1.8 nan]
 [6.  3.4 4.5 1.6 nan]
 [6.7 3.1 5.6 2.4 nan]
 [6.9 3.1 5.4 2.1 nan]
 [5.8 2.7 5.1 1.9 nan]
 [7.1 3.  5.9 2.1 nan]
 [6.6 3.  4.4 1.4 nan]
 [7.7 3.  6.1 2.3 nan]
 [6.3 2.9 5.6 1.8 nan]
 [6.5 3.  5.8 2.2 nan]
 [7.6 3.  6.6 2.1 nan]
 [6.9 3.1 5.4 2.1 nan]
 [5.6 2.8 4.9 2.  nan]
 [6.7 3.  5.2 2.3 nan]
 [7.2 3.2 6.  1.8 nan]
 [6.5 3.  5.5 1.8 nan]
 [6.4 2.7 5.3 1.9 nan]
 [6.8 3.  5.5 2.1 nan]
 [5.8 2.7 5.1 1.9 nan]
 [6.4 3.2 5.3 2.3 nan]
 [6.5 3.  5.8 2.2 nan]
 [6.9 3.1 5.4 2.1 nan]
 [7.  3.2 4.7 1.4 nan]
 [6.4 3.2 4.5 1.5 nan]
 [6.9 3.1 4.9 1.5 nan]
 [5.5 2.3 4.  1.3 nan]
 [6.5 2.8 4.6 1.5 nan]
 [5.7 2.8 4.5 1.3 nan]
 [6.3 3.3 4.7 1.6 nan]
 [4.9 2.4 3.3 1.  nan]
 [6.6 2.9 4.6 1.3 nan]
 [5.2 2.7 3.9 1.4 nan]
 [5.  2.  3.5 1.  nan]
 [5.9 3.  4.2 1.5 nan]
 [6.  2.2 4.  1.  nan]
 [6.1 2.9 4.7 1.4 nan]
 [5.6 2.9 3.6 1.3 nan]
 [6.7 3.1 4.4 1.4 nan]
 [5.6 3.  4.5 1.5 nan]
 [5.8 2.7 4.1 1.  nan]
 [6.2 2.2 4.5 1.5 nan]
 [5.6 2.5 3.9 1.1 nan]
 [5.9 3.2 4.8 1.8 nan]
 [6.1 2.8 4.  1.3 nan]
 [6.3 2.5 4.9 1.5 nan]
 [6.1 2.8 4.7 1.2 nan]
 [6.4 2.9 4.3 1.3 nan]
 [6.6 3.  4.4 1.4 nan]
 [6.8 2.8 4.8 1.4 nan]
 [6.7 3.  5.  1.7 nan]
 [6.  2.9 4.5 1.5 nan]
 [5.7 2.6 3.5 1.  nan]
 [5.5 2.4 3.8 1.1 nan]
 [5.5 2.4 3.7 1.  nan]
 [5.8 2.7 3.9 1.2 nan]
 [6.  2.7 5.1 1.6 nan]
 [5.4 3.  4.5 1.5 nan]
 [6.  3.4 4.5 1.6 nan]
 [6.7 3.1 4.7 1.5 nan]
 [6.3 2.3 4.4 1.3 nan]
 [5.6 3.  4.1 1.3 nan]
 [5.5 2.5 4.  1.3 nan]
 [5.5 2.6 4.4 1.2 nan]
 [6.1 3.  4.6 1.4 nan]
 [5.8 2.6 4.  1.2 nan]
 [5.  2.3 3.3 1.  nan]
 [5.6 2.7 4.2 1.3 nan]
 [5.7 3.  4.2 1.2 nan]
 [5.7 2.9 4.2 1.3 nan]
 [6.2 2.9 4.3 1.3 nan]
 [5.1 3.5 1.7 0.3 nan]
 [4.9 3.  1.6 0.2 nan]
 [5.  3.3 1.4 0.2 nan]
 [4.7 3.2 1.6 0.2 nan]
 [4.6 3.1 1.5 0.2 nan]
 [5.  3.5 1.6 0.6 nan]
 [5.4 3.7 1.5 0.4 nan]
 [4.6 3.6 1.8 0.4 nan]
 [5.  3.2 1.4 0.2 nan]
 [4.4 3.  1.3 0.2 nan]
 [4.9 3.1 1.5 0.1 nan]
 [5.4 3.4 1.7 0.2 nan]
 [4.8 3.4 1.9 0.2 nan]
 [4.8 3.  1.6 0.2 nan]
 [4.3 3.  1.8 0.4 nan]
 [5.8 4.  1.5 0.3 nan]
 [5.7 4.4 1.2 0.3 nan]
 [5.4 3.9 1.3 0.4 nan]
 [5.1 3.5 1.4 0.3 nan]
 [5.7 3.8 1.7 0.3 nan]
 [5.1 3.8 1.5 0.3 nan]
 [5.4 3.4 1.7 0.2 nan]
 [5.1 3.7 1.5 0.4 nan]
 [4.6 3.6 1.1 0.2 nan]
 [5.1 3.3 1.7 0.5 nan]
 [4.8 3.4 1.5 0.2 nan]
 [5.  3.  1.6 0.2 nan]
 [5.  3.4 1.6 0.4 nan]
 [5.2 3.5 1.5 0.2 nan]
 [5.2 3.4 1.4 0.2 nan]
 [4.7 3.2 1.6 0.2 nan]
 [4.8 3.1 1.6 0.2 nan]
 [5.4 3.4 1.5 0.4 nan]
 [5.2 4.1 1.5 0.1 nan]
 [5.5 4.2 1.4 0.2 nan]
 [4.9 3.1 1.5 0.1 nan]
 [5.  3.2 1.7 0.4 nan]
 [5.5 3.5 1.6 0.6 nan]
 [4.9 3.1 1.5 0.5 nan]]
```

## Тип переменной и форма (shape)

```
print ( "Data type : ", type(data) )
print ( "Data shape : ", data.shape )
print ( data[-4:] )
```

```
Data type :  <class 'numpy.ndarray'>
Data shape :  (128, 5)
[[4.9 3.1 1.5 0.1 nan]
 [5.  3.2 1.7 0.4 nan]
 [5.5 3.5 1.6 0.6 nan]
 [4.9 3.1 1.5 0.5 nan]]
```

## Получение типа набора данных, строки, элемента

```
data1 = np.genfromtxt(data_path, delimiter=",", dtype=None)
print('Shape of the dataset:', data1.shape)
print('Dataset type:', type(data1))
print('A single row of the dataset is type of:', type(data1[0]))
print('Types of elements:', type(data1[0][1]), type(data1[0][4]))
print('Dataset:')
print(data1)
```

```
Shape of the dataset: (128,)
Dataset type: <class 'numpy.ndarray'>
A single row of the dataset is type of: <class 'numpy.void'>
Types of elements: <class 'numpy.float64'> <class 'numpy.bytes_'>
Dataset:
[(7.2, 3.6, 6.1, 2.5, b'Continental') (6.3, 3.3, 6. , 2.5,
b'Continental')
 (6.7, 3. , 5.2, 2.3, b'Continental') (6.5, 3. , 5.8, 2.2,
b'Continental')
 (7.1, 3. , 5.9, 2.1, b'Continental') (6.6, 3. , 4.9, 1.8,
b'Continental')
 (7.3, 2.9, 6.3, 1.8, b'Continental') (6.7, 2.5, 5.8, 1.8,
b'Continental')
 (7.2, 3.6, 6.1, 2.5, b'Continental') (6.3, 2.5, 5. , 1.9,
b'Continental')
 (7.2, 3.2, 6. , 1.8, b'Continental') (6.2, 3.4, 5.4, 2.3,
b'Continental')
 (7. , 3.2, 6. , 1.8, b'Continental') (6.9, 3.1, 5.1, 2.3,
b'Continental')
 (6.4, 2.7, 5.3, 1.9, b'Continental') (6.2, 2.2, 4.5, 1.5,
b'Continental')
 (6.8, 3. , 5.5, 2.1, b'Continental') (6. , 2.7, 5.1, 1.6,
b'Continental')
 (5.6, 2.5, 3.9, 1.1, b'Continental') (6.5, 3. , 5.5, 1.8,
b'Continental')
 (6. , 3.4, 4.5, 1.6, b'Continental') (6.7, 3.1, 5.6, 2.4,
b'Continental')
 (6.9, 3.1, 5.4, 2.1, b'Continental') (5.8, 2.7, 5.1, 1.9,
b'Continental')
 (7.1, 3. , 5.9, 2.1, b'Continental') (6.6, 3. , 4.4, 1.4,
b'Continental')
 (7.7, 3. , 6.1, 2.3, b'Continental') (6.3, 2.9, 5.6, 1.8,
b'Continental')
 (6.5, 3. , 5.8, 2.2, b'Continental') (7.6, 3. , 6.6, 2.1,
b'Continental')
 (6.9, 3.1, 5.4, 2.1, b'Continental') (5.6, 2.8, 4.9, 2. ,
b'Continental')
 (6.7, 3. , 5.2, 2.3, b'Continental') (7.2, 3.2, 6. , 1.8,
b'Continental')
 (6.5, 3. , 5.5, 1.8, b'Continental') (6.4, 2.7, 5.3, 1.9,
b'Continental')
 (6.8, 3. , 5.5, 2.1, b'Continental') (5.8, 2.7, 5.1, 1.9,
b'Continental')
 (6.4, 3.2, 5.3, 2.3, b'Continental') (6.5, 3. , 5.8, 2.2,
b'Continental')
 (6.9, 3.1, 5.4, 2.1, b'Continental') (7. , 3.2, 4.7, 1.4,
b'Goodyear')
 (6.4, 3.2, 4.5, 1.5, b'Goodyear') (6.9, 3.1, 4.9, 1.5, b'Goodyear')
 (5.5, 2.3, 4. , 1.3, b'Goodyear') (6.5, 2.8, 4.6, 1.5, b'Goodyear')
 (5.7, 2.8, 4.5, 1.3, b'Goodyear') (6.3, 3.3, 4.7, 1.6, b'Goodyear')
 (4.9, 2.4, 3.3, 1. , b'Goodyear') (6.6, 2.9, 4.6, 1.3, b'Goodyear')
 (5.2, 2.7, 3.9, 1.4, b'Goodyear') (5. , 2. , 3.5, 1. , b'Goodyear')
 (5.9, 3. , 4.2, 1.5, b'Goodyear') (6. , 2.2, 4. , 1. , b'Goodyear')
 (6.1, 2.9, 4.7, 1.4, b'Goodyear') (5.6, 2.9, 3.6, 1.3, b'Goodyear')
 (6.7, 3.1, 4.4, 1.4, b'Goodyear') (5.6, 3. , 4.5, 1.5, b'Goodyear')
 (5.8, 2.7, 4.1, 1. , b'Goodyear') (6.2, 2.2, 4.5, 1.5, b'Goodyear')
 (5.6, 2.5, 3.9, 1.1, b'Goodyear') (5.9, 3.2, 4.8, 1.8, b'Goodyear')
 (6.1, 2.8, 4. , 1.3, b'Goodyear') (6.3, 2.5, 4.9, 1.5, b'Goodyear')
 (6.1, 2.8, 4.7, 1.2, b'Goodyear') (6.4, 2.9, 4.3, 1.3, b'Goodyear')
 (6.6, 3. , 4.4, 1.4, b'Goodyear') (6.8, 2.8, 4.8, 1.4, b'Goodyear')
 (6.7, 3. , 5. , 1.7, b'Goodyear') (6. , 2.9, 4.5, 1.5, b'Goodyear')
 (5.7, 2.6, 3.5, 1. , b'Goodyear') (5.5, 2.4, 3.8, 1.1, b'Goodyear')
 (5.5, 2.4, 3.7, 1. , b'Goodyear') (5.8, 2.7, 3.9, 1.2, b'Goodyear')
 (6. , 2.7, 5.1, 1.6, b'Goodyear') (5.4, 3. , 4.5, 1.5, b'Goodyear')
 (6. , 3.4, 4.5, 1.6, b'Goodyear') (6.7, 3.1, 4.7, 1.5, b'Goodyear')
 (6.3, 2.3, 4.4, 1.3, b'Goodyear') (5.6, 3. , 4.1, 1.3, b'Goodyear')
 (5.5, 2.5, 4. , 1.3, b'Goodyear') (5.5, 2.6, 4.4, 1.2, b'Goodyear')
 (6.1, 3. , 4.6, 1.4, b'Goodyear') (5.8, 2.6, 4. , 1.2, b'Goodyear')
 (5. , 2.3, 3.3, 1. , b'Goodyear') (5.6, 2.7, 4.2, 1.3, b'Goodyear')
 (5.7, 3. , 4.2, 1.2, b'Goodyear') (5.7, 2.9, 4.2, 1.3, b'Goodyear')
 (6.2, 2.9, 4.3, 1.3, b'Goodyear') (5.1, 3.5, 1.7, 0.3, b'Falken')
 (4.9, 3. , 1.6, 0.2, b'Falken') (5. , 3.3, 1.4, 0.2, b'Falken')
 (4.7, 3.2, 1.6, 0.2, b'Falken') (4.6, 3.1, 1.5, 0.2, b'Falken')
 (5. , 3.5, 1.6, 0.6, b'Falken') (5.4, 3.7, 1.5, 0.4, b'Falken')
 (4.6, 3.6, 1.8, 0.4, b'Falken') (5. , 3.2, 1.4, 0.2, b'Falken')
 (4.4, 3. , 1.3, 0.2, b'Falken') (4.9, 3.1, 1.5, 0.1, b'Falken')
 (5.4, 3.4, 1.7, 0.2, b'Falken') (4.8, 3.4, 1.9, 0.2, b'Falken')
 (4.8, 3. , 1.6, 0.2, b'Falken') (4.3, 3. , 1.8, 0.4, b'Falken')
 (5.8, 4. , 1.5, 0.3, b'Falken') (5.7, 4.4, 1.2, 0.3, b'Falken')
 (5.4, 3.9, 1.3, 0.4, b'Falken') (5.1, 3.5, 1.4, 0.3, b'Falken')
 (5.7, 3.8, 1.7, 0.3, b'Falken') (5.1, 3.8, 1.5, 0.3, b'Falken')
 (5.4, 3.4, 1.7, 0.2, b'Falken') (5.1, 3.7, 1.5, 0.4, b'Falken')
 (4.6, 3.6, 1.1, 0.2, b'Falken') (5.1, 3.3, 1.7, 0.5, b'Falken')
 (4.8, 3.4, 1.5, 0.2, b'Falken') (5. , 3. , 1.6, 0.2, b'Falken')
 (5. , 3.4, 1.6, 0.4, b'Falken') (5.2, 3.5, 1.5, 0.2, b'Falken')
 (5.2, 3.4, 1.4, 0.2, b'Falken') (4.7, 3.2, 1.6, 0.2, b'Falken')
 (4.8, 3.1, 1.6, 0.2, b'Falken') (5.4, 3.4, 1.5, 0.4, b'Falken')
 (5.2, 4.1, 1.5, 0.1, b'Falken') (5.5, 4.2, 1.4, 0.2, b'Falken')
 (4.9, 3.1, 1.5, 0.1, b'Falken') (5. , 3.2, 1.7, 0.4, b'Falken')
 (5.5, 3.5, 1.6, 0.6, b'Falken') (4.9, 3.1, 1.5, 0.5, b'Falken')]

<ipython-input-5-be1180784c4e>:1: VisibleDeprecationWarning: Reading
unicode strings without specifying the encoding argument is
deprecated. Set the encoding, use None for the system default.
  data1 = np.genfromtxt(data_path, delimiter=",", dtype=None)
```

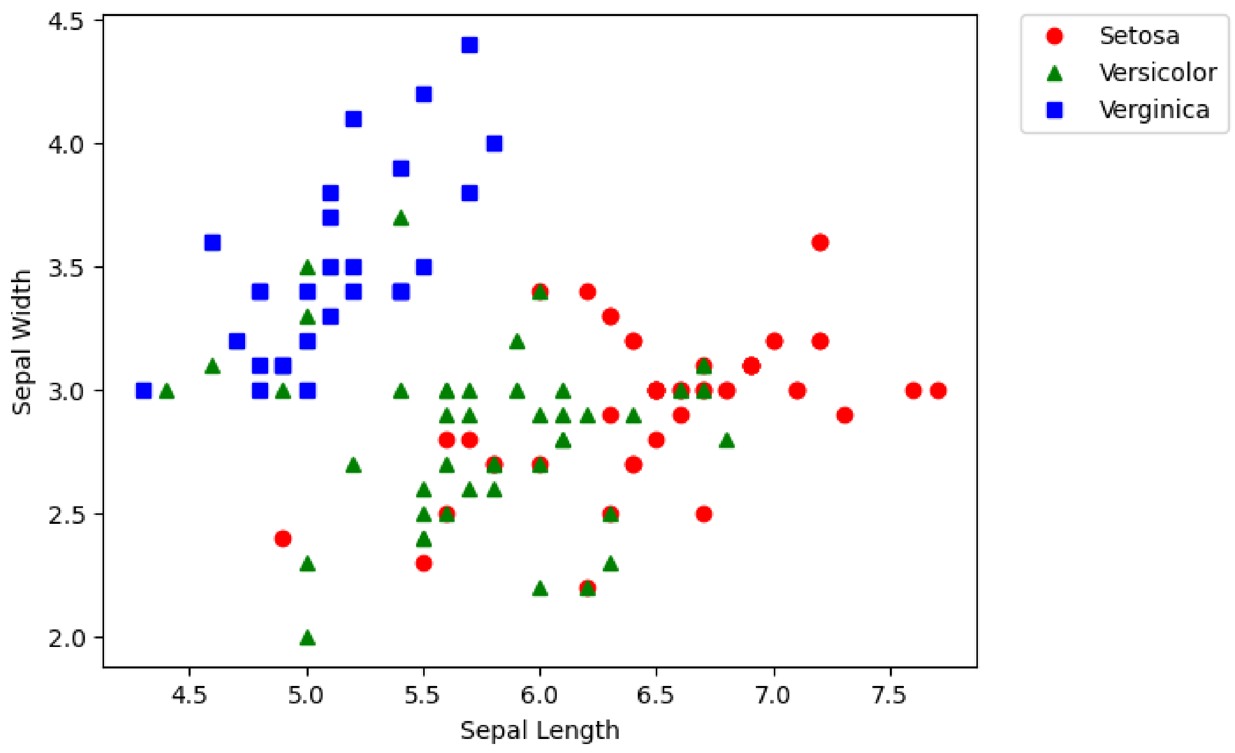
## Указание типа столбцов при загрузке данных

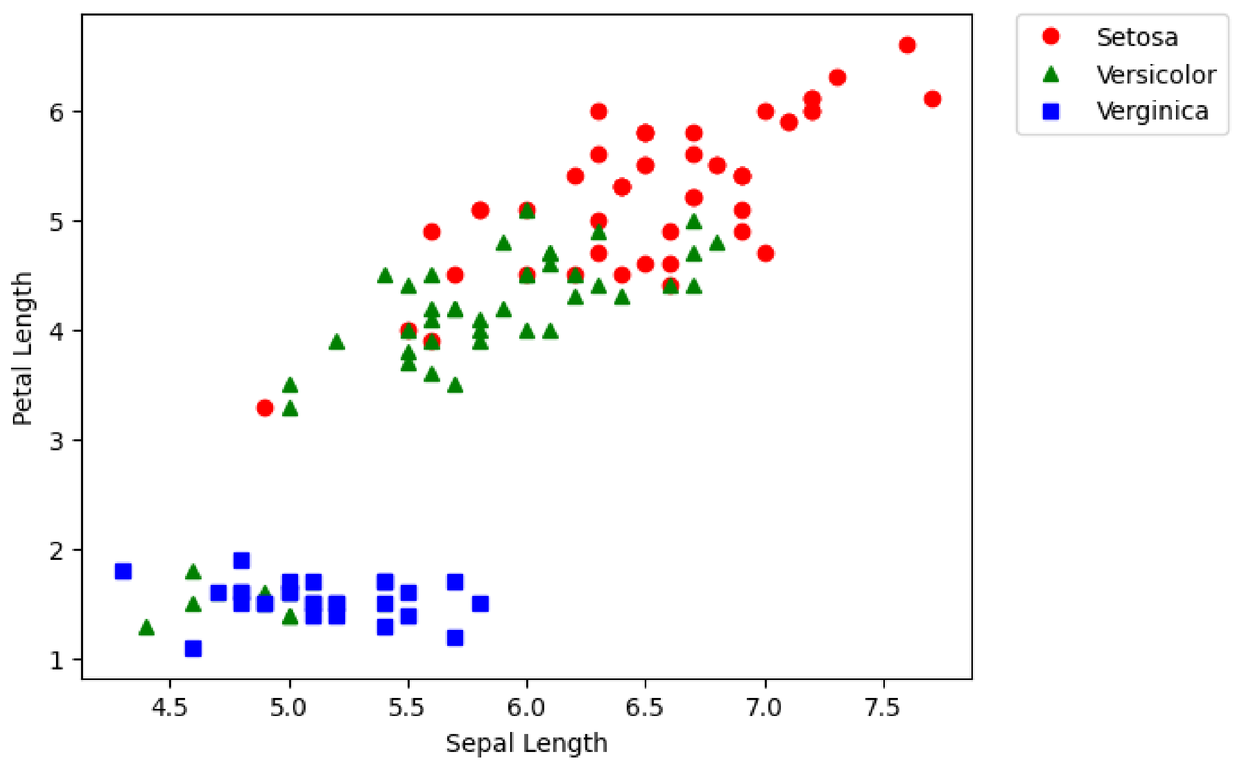
```
dt = np.dtype("f8, f8, f8, f8, U30")
data2 = np.genfromtxt(data_path, delimiter=",", dtype=dt)
print('Shape of the dataset:', data2.shape)
print('Dataset type:', type(data2))
print('A single row of the dataset is type of:', type(data2[0]))
print('Types of elements:', type(data2[0][1]), type(data2[0][4]))
print('Dataset slice:')
print(data2[:10])

Shape of the dataset: (128,)
Dataset type: <class 'numpy.ndarray'>
```

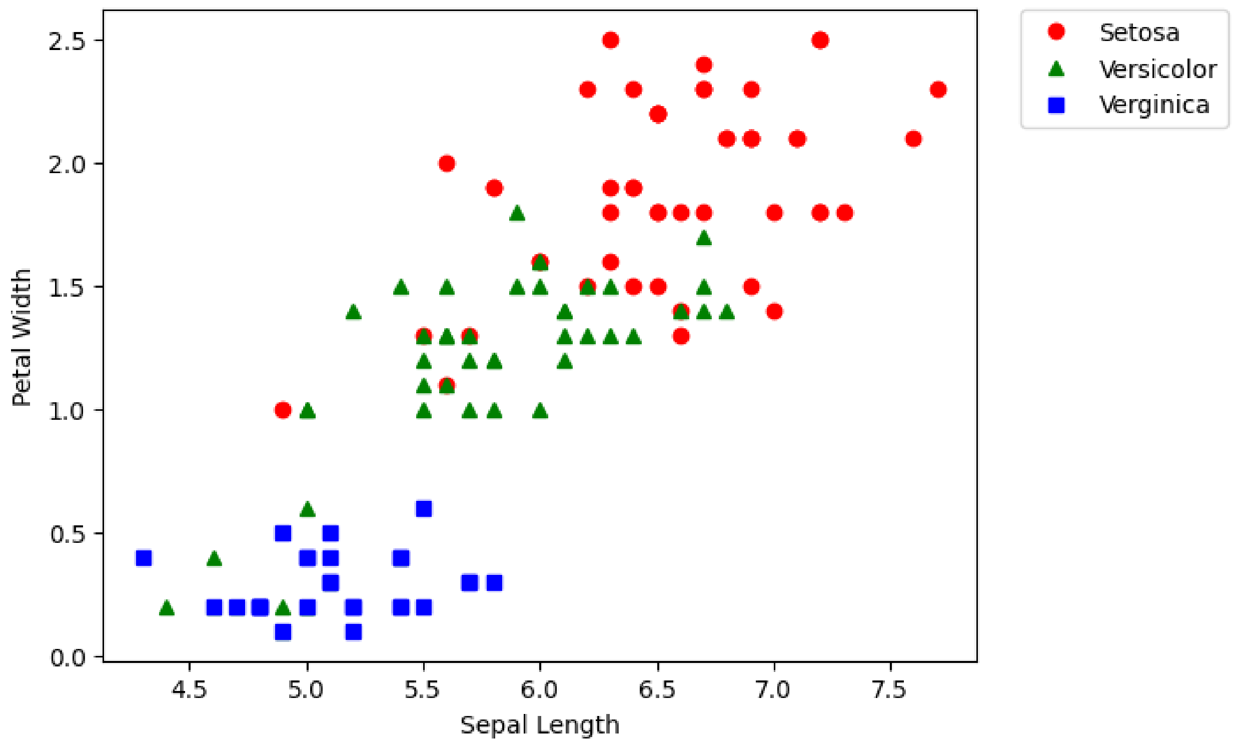
```
A single row of the dataset is type of: <class 'numpy.void'>
Types of elements: <class 'numpy.float64'> <class 'numpy.str_'>
Dataset slice:
[(7.2, 3.6, 6.1, 2.5, 'Continental') (6.3, 3.3, 6. , 2.5,
'Continental')
 (6.7, 3. , 5.2, 2.3, 'Continental') (6.5, 3. , 5.8, 2.2,
'Continental')
 (7.1, 3. , 5.9, 2.1, 'Continental') (6.6, 3. , 4.9, 1.8,
'Continental')
 (7.3, 2.9, 6.3, 1.8, 'Continental') (6.7, 2.5, 5.8, 1.8,
'Continental')
 (7.2, 3.6, 6.1, 2.5, 'Continental') (6.3, 2.5, 5. , 1.9,
'Continental')]
```

## Построение графиков с использованием Matplotlib

```python
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline

# Данные из отдельных столбцов
sepal_length = [] # Sepal Length
sepal_width = [] # Sepal Width
petal_length = [] # Petal Length
petal_width = [] # Petal Width

# Выполняем обход всей коллекции data2
for dot in data2:
    sepal_length.append(dot[0])
    sepal_width.append(dot[1])
    petal_length.append(dot[2])
    petal_width.append(dot[3])

# Строим графики по проекциям данных
# Учитываем, что каждые 50 типов ирисов идут последовательно
plt.figure(1)
setosa, = plt.plot(sepal_length[:50], sepal_width[:50], 'ro',
label='Setosa')
versicolor, = plt.plot(sepal_length[50:100], sepal_width[50:100],
'g^', label='Versicolor')
virginica, = plt.plot(sepal_length[100:150], sepal_width[100:150],
'bs', label='Verginica')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel('Sepal Length')
plt.ylabel('Sepal Width')

plt.figure(2)
setosa, = plt.plot(sepal_length[:50], petal_length[:50], 'ro',
label='Setosa')
versicolor, = plt.plot(sepal_length[50:100], petal_length[50:100],
'g^', label='Versicolor')
virginica, = plt.plot(sepal_length[100:150], petal_length[100:150],
'bs', label='Verginica')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel('Sepal Length')
plt.ylabel('Petal Length')

plt.figure(3)
setosa, = plt.plot(sepal_length[:50], petal_width[:50], 'ro',
label='Setosa')
versicolor, = plt.plot(sepal_length[50:100], petal_width[50:100],
'g^', label='Versicolor')
virginica, = plt.plot(sepal_length[100:150], petal_width[100:150],
'bs', label='Verginica')
plt.legend(bbox_to_anchor=(1.05, 1), loc=2, borderaxespad=0.)
plt.xlabel('Sepal Length')
plt.ylabel('Petal Width')

plt.show()
```




Setosa
Versicolor
Verginica
Petal Length
Sepal Length
1
2
3
4
5
6
4.5
5.0
5.5
6.0
6.5
7.0
7.5


Setosa
Versicolor
Verginica
Petal Width
Sepal Length
0.0
0.5
1.0
1.5
2.0
2.5
4.5
5.0
5.5
6.0
6.5
7.0
7.5