

Practical No. 2

Title: Introduction to Python Programming and python libraries

Aim: Understanding basics of python programming and python libraries like numpy, pandas and matplotlib

Introduction:

What is Python :-

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python libraries for Machine Learning

Machine Learning, as the name suggests, is the science of programming a computer by which they are able to learn from different kinds of data. A more general definition given by Arthur Samuel is – “Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.” They are typically used to solve various types of life problems.

In the older days, people used to perform Machine Learning tasks by manually coding all the algorithms and mathematical and statistical formulas. This made the processing time consuming, tedious and inefficient. But in the modern days, it has become very much easy and efficient compared to the olden days with various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reasons is its vast collection of libraries. Python libraries that used in Machine Learning are:

Numpy

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow use NumPy internally for manipulation of Tensors.

Pandas

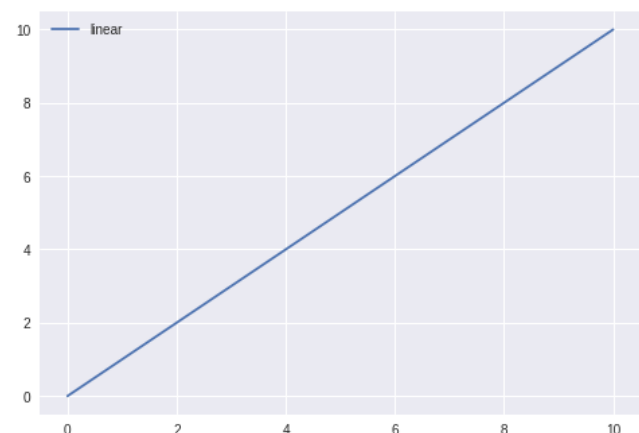
Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and a wide variety of tools for data analysis. It provides many inbuilt methods for grouping, combining and filtering data.

Matplotlib

Matplotlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar charts, etc,

Example :-

```
import matplotlib.pyplot as plt
import numpy as np
x = np.linspace(0, 10, 100)
plt.plot(x, x, label = 'linear')
plt.legend()
plt.show()
```



Exercise -**Implementation:**

Create Pandas Dataframe to enter employee database which includes Sr No, Name, Mobile No, City

Program:

```
import pandas as pd
```

```
# Sample employee data with Indian names
```

```
employee_data = [
```

```
    {"Sr No": 1, "Name": "Onkar", "Mobile No": "9876543210", "City": "Mumbai"},
```

```
    {"Sr No": 2, "Name": "Paresh", "Mobile No": "9876543211", "City": "Delhi"},
```

```
    {"Sr No": 3, "Name": "Dhruv", "Mobile No": "9876543212", "City": "Bangalore"},
```

```
    {"Sr No": 4, "Name": "Dattu", "Mobile No": "9876543213", "City": "Ahmedabad"},
```

```
    # Add more employee data as needed
```

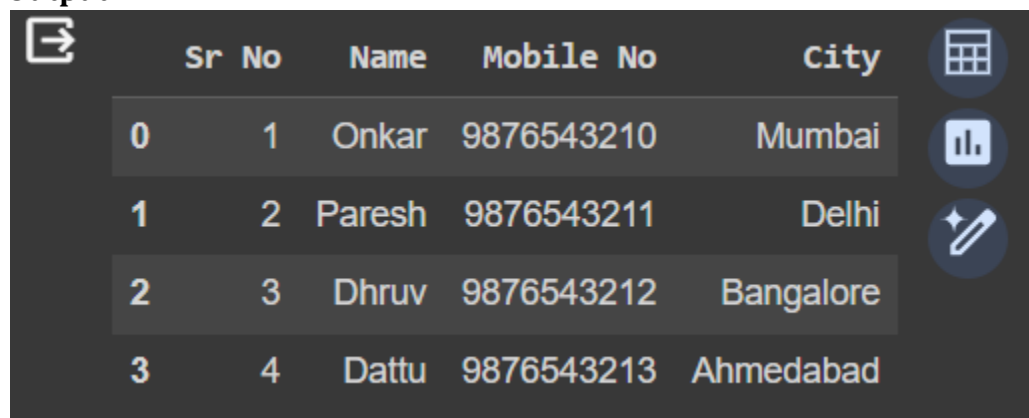
```
]
```

```
# Create DataFrame
```

```
employee_df = pd.DataFrame(employee_data)
```

```
# Display DataFrame
```

```
employee_df
```

Output:

	Sr No	Name	Mobile No	city
0	1	Onkar	9876543210	Mumbai
1	2	Paresh	9876543211	Delhi
2	3	Dhruv	9876543212	Bangalore
3	4	Dattu	9876543213	Ahmedabad

Use Iris Dataset from Github and perform all basic operations on Iris Dataset**Program:**

```
import pandas as pd
```

```
url =
```

```
'https://gist.githubusercontent.com/curran/a08a1080b88344b0c8a7/raw/0e7a9b0a5d2  
2642a06d3d5b9bcbad9890c8ee534/iris.csv'
```

```
iris = pd.read_csv(url)
```

```
iris
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

```
In [4]: iris.head()
```

```
Out[4]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

```
In [6]: iris.iloc[75:77]
```

```
Out[6]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
75	6.6	3.0	4.4	1.4	versicolor
76	6.8	2.8	4.8	1.4	versicolor

```
In [7]: iris.loc[74:79,"sepal_width"]
```

```
Out[7]:
```

74	2.9
75	3.0
76	2.8
77	3.0
78	2.9
79	2.6

Name: sepal_width, dtype: float64

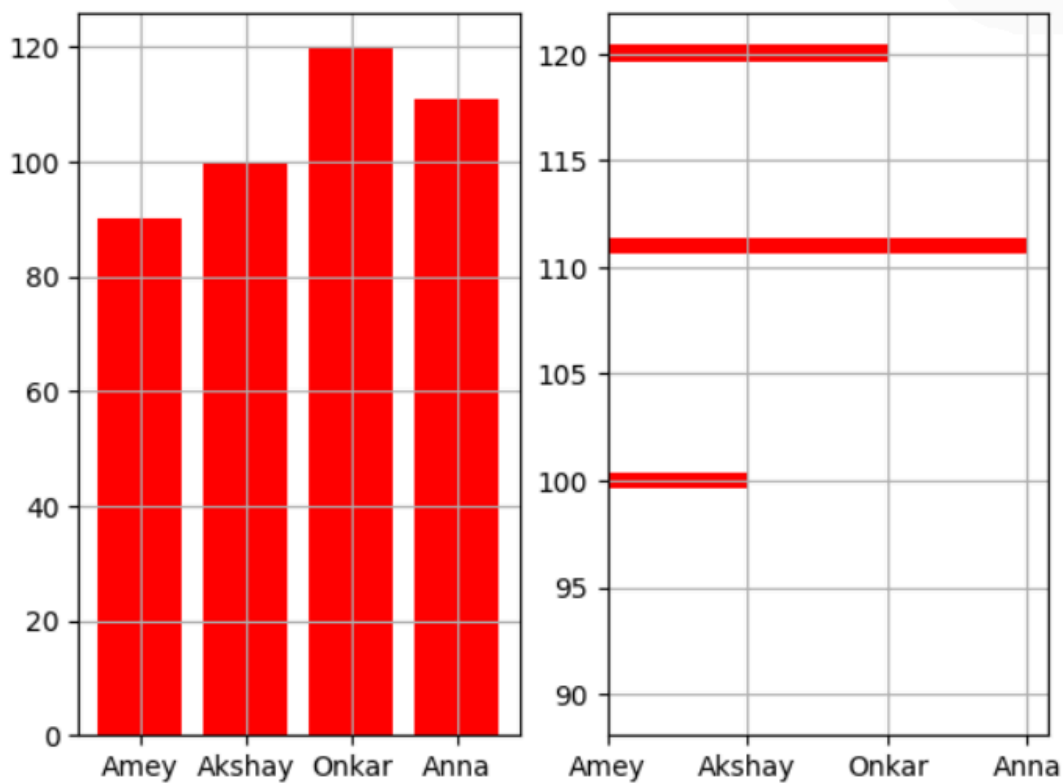
```
In [8]: iris.drop([0,1,2],axis = 0)
```

```
Out[8]:
```

	sepal_length	sepal_width	petal_length	petal_width	species
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

Use matplotlib to plot bar chart using dictionary**Program :**

```
student={"Amey":90,"Akshay":100,"Onkar":120,"Anna":111}  
name = list(student.keys())  
marks = list(student.values())  
  
plt.subplot(1,2,1)  
plt.bar(name,marks,color='r')  
plt.grid(True)  
  
plt.subplot(1,2,2)  
plt.barh(marks,name,color='r')  
plt.grid(True)  
  
plt.show()
```



Use matplotlib to scatter graph and histogram**Program:**

```
x = np.array([1,2,3,4,5,6,7,8])  
y = np.array([10,20,30,40,50,60,70,80])  
  
plt.scatter(x,y,color='y',marker="*")  
plt.show()
```

Output: