Nanditha.V
160119733133

Data Science and Big Data Analytics

Experiment 1: Installation of required software, Programs using Numpy, Pandas and Matplotlib

AIM: Identification and Installation of required software and technology (python modules)

DESCRIPTION:
The jupyter notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document centric experience.

Libraries for python:

- Tensorflow
- Numpy
- Scipy
- Pandas
- Matplotlib
- Keras
- Scikit-learn
- Pytorch
- Scrapy
- BeautifulSoup

Numpy:
NumPy is a Python library used for working with arrays.It also has functions for working in domain of linear algebra, fourier transform, and matrices.NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.NumPy stands for Numerical Python.

Scipy:
SciPy is a scientific computation library that uses NumPy underneath.SciPy stands for Scientific Python.It provides more utility functions for optimization, stats and signal processing.Like NumPy, SciPy is open source so we can use it freely.SciPy was created by NumPy's creator Travis Olliphant.If SciPy uses NumPy underneath, why can we not just use NumPy?SciPy has optimized and added functions that are frequently used in NumPy and Data Science.

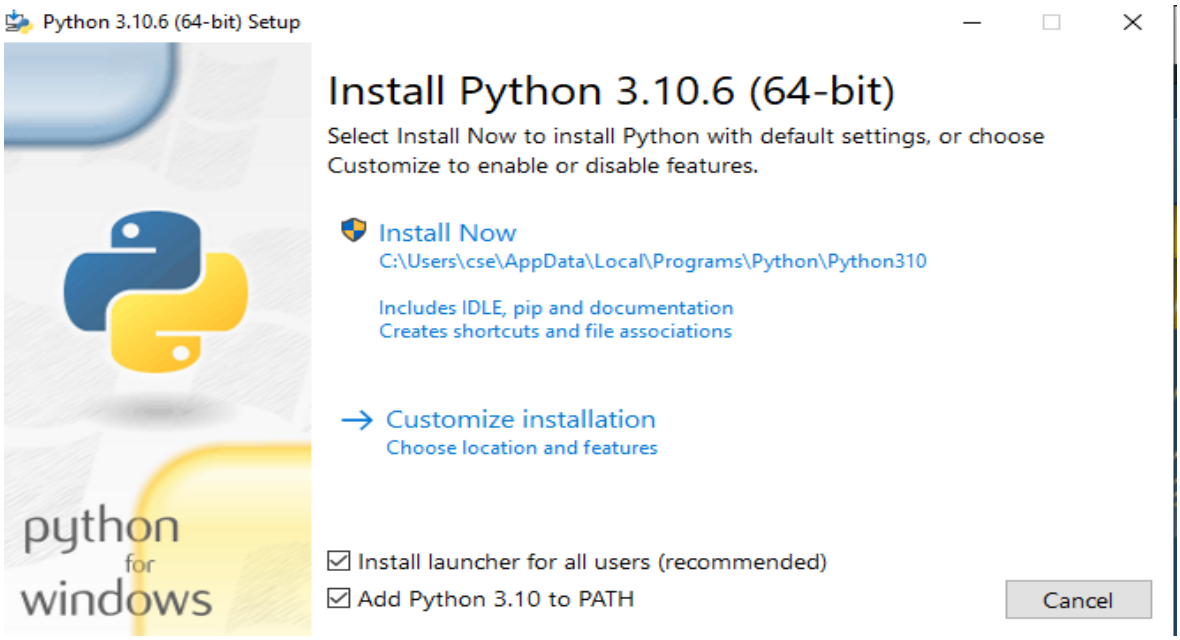
Pandas:
Pandas is a Python library used for working with data sets.It has functions for analyzing, cleaning, exploring, and manipulating data.The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.Pandas allows us to analyze big data and make conclusions based on statistical theories.Pandas can clean messy data sets, and make them readable and relevant.Relevant data is very important in data science.

Matplotlib:
Matplotlib is a low level graph plotting library in python that serves as a visualization utility.Matplotlib was created by John D. Hunter.Matplotlib is open source and we can use it

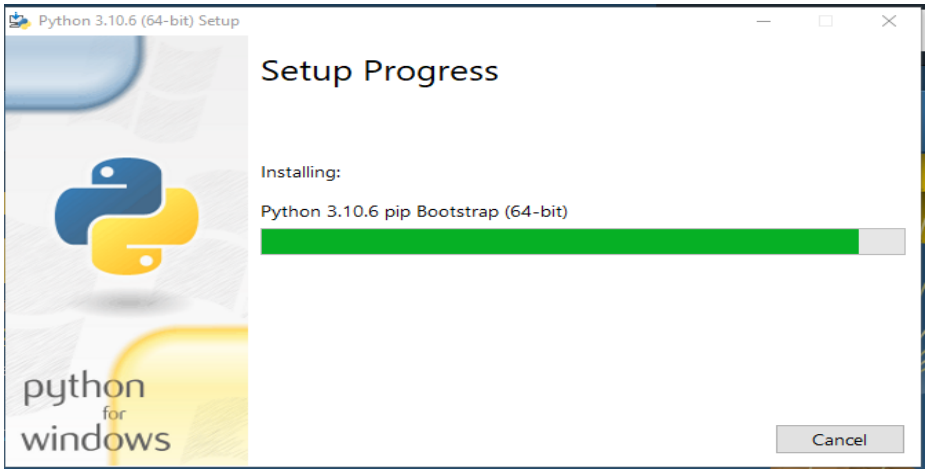
freely.Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility.

PROCEDURE:
Install python:

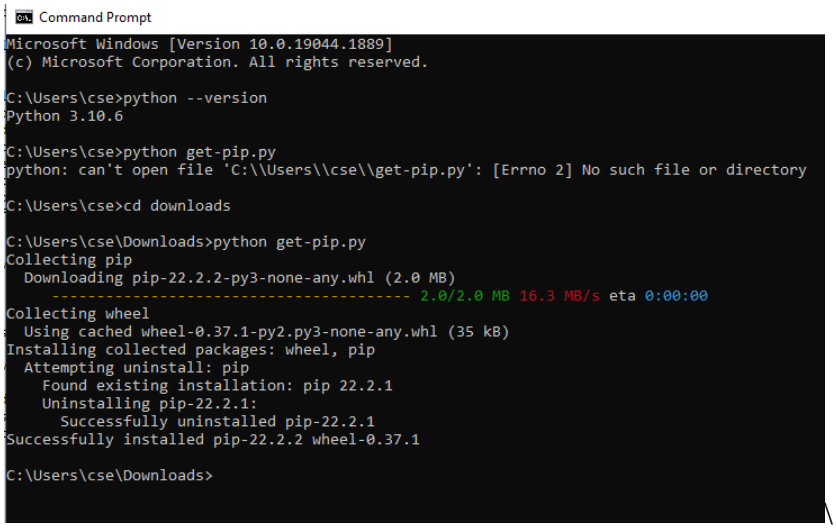
Go to python.org/downloads and download the latest version of python.



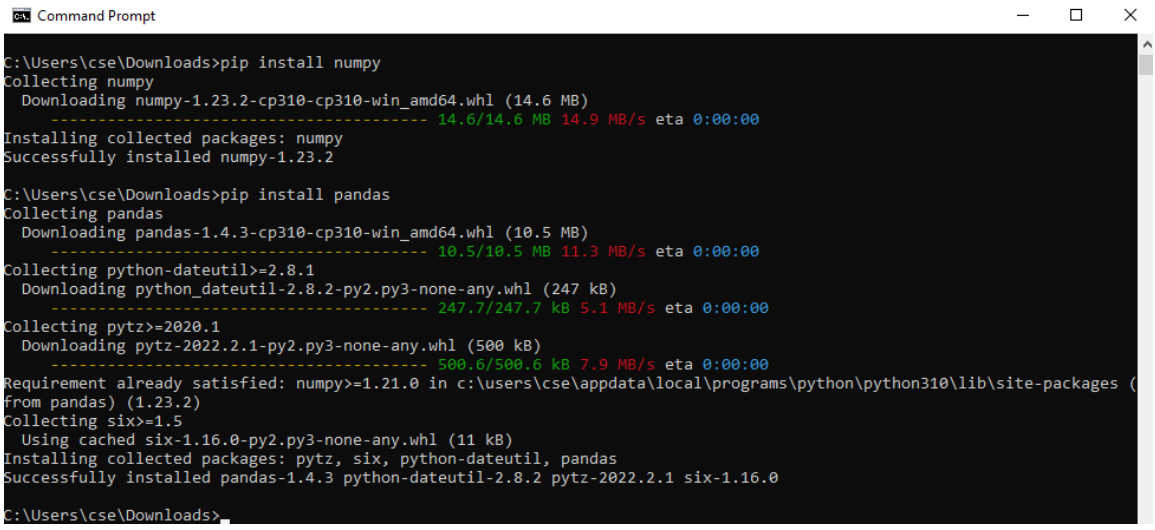
Nanditha.V
160119733133



Install pip:
python get-pip.py

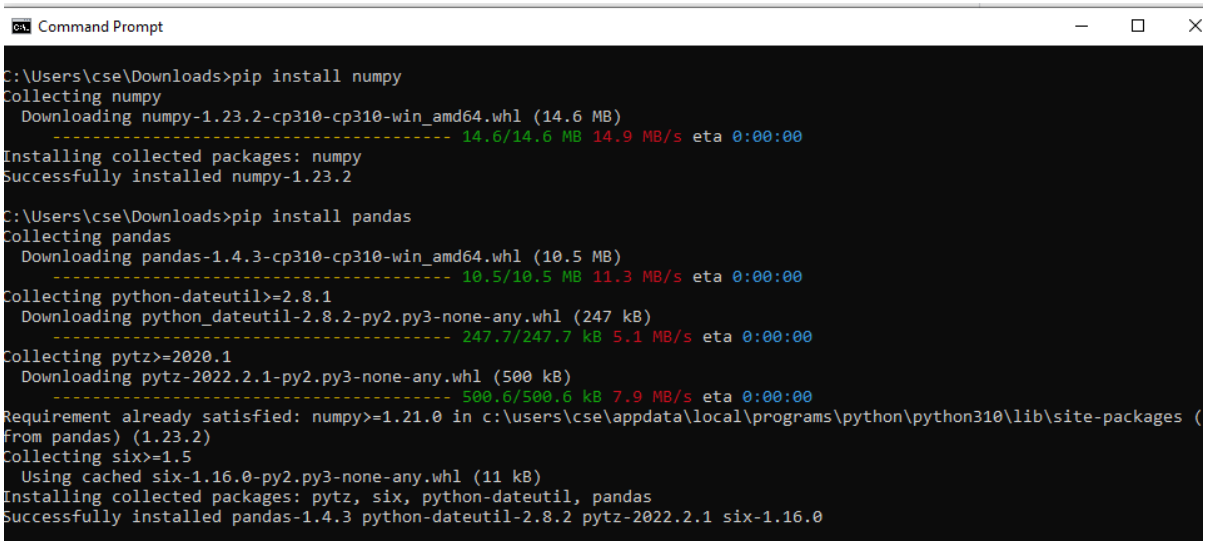


To install Numpy:
pip install numpy

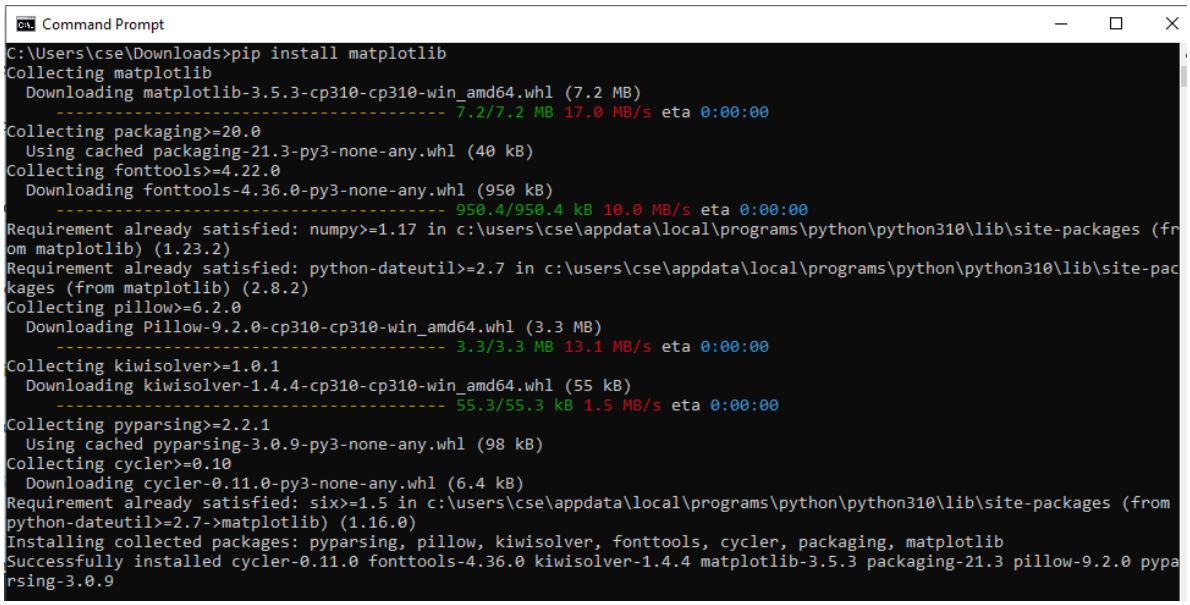


To install Pandas:

pip install pandas



To install Matplotlib:
pip install matplotlib



To install scipy:
pip install scipy

```
Command Prompt
Requirement already satisfied: numpy>=1.17 in c:\users\cse\appdata\local\programs\python\python310\lib\site-packages (fr
om matplotlib) (1.23.2)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\cse\appdata\local\programs\python\python310\lib\site-pac
kages (from matplotlib) (2.8.2)
Collecting pillow>=6.2.0
  Downloading Pillow-9.2.0-cp310-cp310-win_amd64.whl (3.3 MB)
     ---------------------------------------- 3.3/3.3 MB 13.1 MB/s eta 0:00:00
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.4.4-cp310-cp310-win_amd64.whl (55 kB)
     ---------------------------------------- 55.3/55.3 kB 1.5 MB/s eta 0:00:00
Collecting pyparsing>=2.2.1
  Using cached pyparsing-3.0.9-py3-none-any.whl (98 kB)
Collecting cycler>=0.10
  Downloading cycler-0.11.0-py3-none-any.whl (6.4 kB)
Requirement already satisfied: six>=1.5 in c:\users\cse\appdata\local\programs\python\python310\lib\site-packages (from
python-dateutil>=2.7->matplotlib) (1.16.0)
Installing collected packages: pyparsing, pillow, kiwisolver, fonttools, cycler, packaging, matplotlib
Successfully installed cycler-0.11.0 fonttools-4.36.0 kiwisolver-1.4.4 matplotlib-3.5.3 packaging-21.3 pillow-9.2.0 pypa
rsing-3.0.9

C:\Users\cse\Downloads>pip install scipy
Collecting scipy
  Downloading scipy-1.9.0-cp310-cp310-win_amd64.whl (38.6 MB)
     ---------------------------------------- 38.6/38.6 MB 15.2 MB/s eta 0:00:00
Requirement already satisfied: numpy<1.25.0,>=1.18.5 in c:\users\cse\appdata\local\programs\python\python310\lib\site-pa
ckages (from scipy) (1.23.2)
Installing collected packages: scipy
Successfully installed scipy-1.9.0
```

To install jupyter notebook(linux):
sudo snap install jupyter
To install jupyter notebook(windows):
pip install jupyter notebook

```
Command Prompt - pip  install jupyter notebook
Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. All rights reserved.

C:\Users\cse>pip install jupyter notebook
Collecting jupyter
  Downloading jupyter-1.0.0-py2.py3-none-any.whl (2.7 kB)
Collecting notebook
  Downloading notebook-6.4.12-py3-none-any.whl (9.9 MB)
     ---------------------------------------- 9.9/9.9 MB 17.6 MB/s eta 0:00:00
Collecting nbconvert
  Downloading nbconvert-6.5.3-py3-none-any.whl (563 kB)
     ---------------------------------------- 563.8/563.8 kB 5.9 MB/s eta 0:00:00
Collecting jupyter-console
  Downloading jupyter_console-6.4.4-py3-none-any.whl (22 kB)
Collecting qtconsole
  Downloading qtconsole-5.3.1-py3-none-any.whl (120 kB)
     ---------------------------------------- 120.8/120.8 kB 1.8 MB/s eta 0:00:00
Collecting ipykernel
  Downloading ipykernel-6.15.1-py3-none-any.whl (132 kB)
     ---------------------------------------- 132.9/132.9 kB 7.7 MB/s eta 0:00:00
Collecting ipywidgets
  Downloading ipywidgets-7.7.1-py2.py3-none-any.whl (123 kB)
     ---------------------------------------- 123.4/123.4 kB 3.6 MB/s eta 0:00:00
Collecting argon2-cffi
  Downloading argon2_cffi-21.3.0-py3-none-any.whl (14 kB)
Collecting Send2Trash>=1.8.0
  Downloading Send2Trash-1.8.0-py3-none-any.whl (18 kB)
Collecting nest-asyncio>=1.5
  Downloading nest_asyncio-1.5.5-py3-none-any.whl (5.2 kB)
Collecting pyzmq>=17
```

```
C:\Users\cse>python3
Python 3.9.7 (tags/v3.9.7:1016ef3, Aug 30 2021, 20:19:38) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>> import scipy
>>> import matplotlib
>>> import pandas
>>>
```

Nanditha.V
160119733133

PROGRAMS USING THE INSTALLED MODULES:

A) Creating and printing a one-dimensional array

```
In [2]: import numpy
        arr=numpy.array([4,8,12])
        arr
Out[2]: array([ 4,  8, 12])

In [3]: type(arr)
Out[3]: numpy.ndarray

In [4]: arr.dtype
Out[4]: dtype('int32')

In [ ]:
```

B) Creating and printing a two-dimensional array

```
In [6]: arr=numpy.array([(1.1,2,3,4),(5,6,7,8)])
        arr
Out[6]: array([[1.1, 2. , 3. , 4. ],
               [5. , 6. , 7. , 8. ]])

In [7]: type(arr)
Out[7]: numpy.ndarray

In [9]: arr.dtype
Out[9]: dtype('float64')

In [ ]:

In [ ]:
```

C) Product of a two-dimensional array

```
In [11]: arr=numpy.array([(1,3,1),(2,2,2)])
         numpy.product(arr)
Out[11]: 24

In [ ]:
```

Nanditha.V
160119733133

D) Indexing and slicing

```
In [12]: arr=numpy.arange(10)
         arr
Out[12]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [13]: sli=arr[3:10:2]
         sli
Out[13]: array([3, 5, 7, 9])

In [15]: c=arr>2
         d=arr[c]
         d
Out[15]: array([3, 4, 5, 6, 7, 8, 9])

In [ ]:
```

E) Iterating

```
In [15]: c=arr>2
         d=arr[c]
         d
Out[15]: array([3, 4, 5, 6, 7, 8, 9])

In [17]: import numpy as np
         a=np.arange(0,50,5)
         for i in np.nditer(a):
             print(i)

         0
         5
         10
         15
         20
         25
         30
         35
         40
         45

In [ ]:
```

F) Reshaping

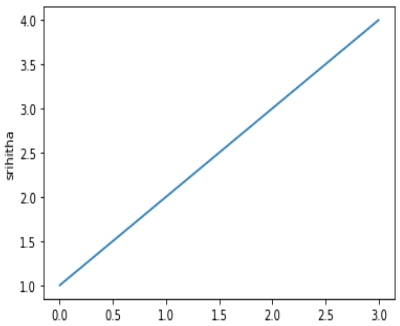
```
In [18]: a=np.arange(0,60,5)
         b=a.reshape(6,2)
         b
Out[18]: array([[ 0,  5],
                [10, 15],
                [20, 25],
                [30, 35],
                [40, 45],
                [50, 55]])

In [ ]:
```

Nanditha.V
160119733133

G) Line plot using Matplotlib

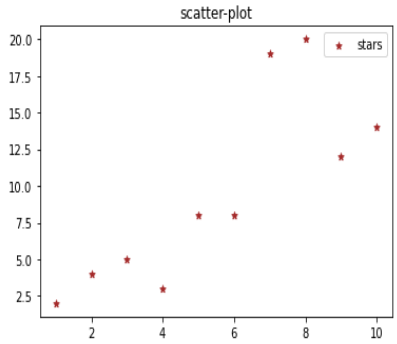
```
In [23]: import matplotlib.pyplot as plt
         plt.plot([1,2,3,4])
         plt.ylabel("srihitha")
         plt.show()
```




```
In [ ]:
```

H) Scatterplot using Matplotlib

```
In [25]: x=[1,2,3,4,5,6,7,8,9,10]
         y=[2,4,5,3,8,8,19,20,12,14]
         plt.scatter(x,y,label="stars",color='brown',marker="*",s=30)
         plt.title('scatter-plot')
         plt.legend()
         plt.show()
```

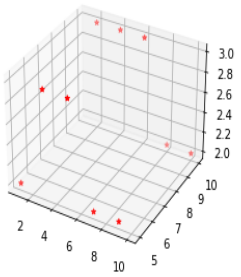



```
In [ ]:
```

Nanditha.V
160119733133

I) 3D Plot using Matplotlib

```
In [28]: from mpl_toolkits.mplot3d import Axes3D
         fig=plt.figure()
         ax=fig.add_subplot(111,projection='3d')
         x=[1,2,3,4,5,6,7,8,9,10]
         y=[5,10,5,10,5,10,5,10,5,10]
         z=[2,3,3,3,3,3,2,2,2,2]
         ax.scatter(x,y,z,c='r',marker='*')
         plt.show()
```



J) DataFrame Implementation using pandas

```
In [29]: import pandas as pd
         lst=['hello','I am','Srihitha']
         df=pd.DataFrame(lst)
         print(df)
```

```
          0
0     hello
1      I am
2  Srihitha
```

```
In [ ]:
```

H) Series Implementation using python

```
In [30]: lst=['hello','I am','Srihitha']
         data=pd.Series(lst)
         print(type(data))
         print(data)
```

```
<class 'pandas.core.series.Series'>
0       hello
1        I am
2    Srihitha
dtype: object
```

```
In [ ]:
```