**ASSIGNMENT – I**

**Installing Jupyter Notebook in our machine:**

<pip install notebook> #run this in command prompt

**Creating a notebook hosted on our local server 8888:**

<jupyter notebook> #run this in command prompt

**Installing Numpy in our machine:**

<pip install numpy> #run this in command prompt

**Installing Pandas in our machine:**

<pip install pandas> #run this in command prompt

**Creating 1D array using Numpy:**

<A = np.array(list1)> #where list1 is elements in row1 in our array

**Creating 2D array using Numpy:**

<A = np.array(list1, list2)> #where list1 is elements in row1 in our array and list2 comprises of elements in row2

**Finding the dimensions of our array:**

<array\_name.shape>

**Finding the datatype of our array:**

<array\_name.dtype> #generally int32 or float64

**Creating array with all elements as 0:**

**Creating 1D array:**

<A = np.zeros(n)> #where n is the number of zeroes in the array, use function ones if you want all elements to be 1

**Creating 2D array:**

<A = np.zeros([n, m])> #where n is number of rows and m is number of columns

**Creating identity array i.e. an array having 1 at i = j and 0 elsewhere:**

<A = np.eye([n, m])> #where n is number of rows and m is number of columns

**Raising each element of an array to an exponent:**

<array\_new = array\_name \*\* n> #where n is the exponential value

**Creating arrays within a given range:**

<array\_name = np.arange(x, y, z)> #creates an array in starting from x till y in the intervals of z but excluding y

**Creating a copy of an array:**

<array\_1 = array\_2.copy()> #copies all the elements of array\_2 to array\_1 without changing any values in array\_2

**Accessing all rows in an array in a particular column:**

<print(array[:, n])> #this will print all the elements in each row of the array in column n

**Accessing all columns in an array in a particular row:**

<print(array[m, :])> #this will print all the elements in each column of the array in row m

**Accessing a particular element in an array:**

<print(array[m, n])> #prints the element in mth row and nth column in the array

**Calculating the square root of each array element:**

<A = np.sqrt(array\_1)>

**Installing matplotlib in our machine for plotting graphs:**

<pip install matplotlib> #to be executed in command prompt

**Importing matplotlib in our notebook**

<import matplotlib.pyplot as plt>

**Simple plotting using matplotlib**

<plt.plot(kind=’kind\_name’, figsize = (n, m))>

<plt.title(‘Plot\_Name’)>

<plt.show()> #to finally display the figure

**Some standard functions in numpy:**

<A = np.sum(array\_name)> #returns sum of all elements in array

<A = np.mean(array\_name)> #returns average of all elements in array

<A = np.std(array\_name)> #returns deviation of all elements in array

**Importing Series, DataFrame from pandas**

<from pandas import Series, DataFrame >

**Creating Series object:**

<A = Series([list], index = [list2])> #a series object is a dataframe with a single column apart from the index column

**Creating DataFrame object:**

<A = pd.DataFrame(dict\_1, dict\_2, dict\_3)>

#loading dictionary values to dataframe where key will become column name while the value will become the entry in the respective column

**Accessing a particular column in a Data Frame:**

<df[‘column\_name’]> #all the values in the column will be retrieved

**Deleting a column from a Data Frame:**

<df.drop(‘column\_name’, axis = 1/0, inplace = True)> #specify the axis as 1 for a column and 0 for a row

**Accessing a particular value in a data frame:**

<df.loc[row\_name, column\_name]>

**Resetting the index in a Data Frame:**

<df.reset\_index()> #will assign numerical indexes to the data frame starting with 0

**Sorting values in a Data Frame:**

<df.sort\_values(by = ‘column\_name’, axis = 1/0, inplace = True)>

#sorts the value in the data frame by the column specified and also the axis while the inplace parameter sets the new changes in our data frame

**Plotting graphs with a Data Frame:**

<df[‘column\_name’].plot(kind=’type’, figsize=(m, n))> #this will plot a graph using the matplotlib library according to the type specified and also the size parameterized by m and n