->pip

-> jupyter notebook

Libraries :

1**. Numpy** : NumPy is a very popular python library for large multi-dimensional array and matrix processing

import numpy as nup

K = nup.array([[2, 4], [6, 8]])

R = nup.array([[1, 3], [5, 7]])

print ("Matrix and matrix product: ", nup.dot(K, R))

2. **Panda :** Pandas is a popular Python library for data analysis. Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis.

import pandas as pd

data = {"country": ["Brazil", "Russia", "India", "China", "South Africa"],

"capital": ["Brasilia", "Moscow", "New Delhi", "Beijing", "Pretoria"],

"area": [8.516, 17.10, 3.286, 9.597, 1.221],

"population": [200.4, 143.5, 1252, 1357, 52.98] }

data\_table = pd.DataFrame(data)

print(data\_table)

Note: Pandas DataFrame is a widely used data structure which works with a two-dimensional array with labeled axes (rows and columns). DataFrame is defined as a standard way to store data that has two different indexes, i.e., **row index** and **column index**.

3. Matplotlib : Matplotlib is a very popular Python library for data visualization.

Matplotlib utilities lies under the pyplot submodule

import matplotlib.pyplot as plt

import numpy as np

xpoints = np.array([0, 6])

ypoints = np.array([0, 250])

plt.plot(xpoints, ypoints)

plt.show()

4.

import pandas as pd

df =pd.read\_csv('vgsales.csv')

df

// The read\_csv() function takes a path to a CSV file and reads the data into a Pandas DataFrame object.

* df.shape ->

5. import pandas as pd

music\_data = pd.read\_csv('music.csv')

X=music\_data.drop(columns=['genre'])

X

6. import pandas as pd

music\_data =pd.read\_csv('music.csv')

X= music\_data.drop(columns=['genre'])

Y=music\_data['genre']

Y

Regression:

The term regression is used when you try to find the relationship between variables.

In Machine Learning, and in statistical modeling, that relationship is used to predict the outcome of future events.

Linear regression:

Linear regression uses the relationship between the data-points to draw a straight line through all them.

This line can be used to predict future values.

7. import matplotlib.pyplot as plt

x = [5,7,8,7,2,17,2,9,4,11,12,9,6]

y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

plt.scatter(x, y)

plt.show()

8. #linear regression

#importing libraries

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

#loading the data

df = pd.read\_csv('placement.csv')

# display the 5 records

df.head()

# plotting the graph

plt.scatter(df['cgpa'],df['package'])

plt.xlabel('CGPA')

plt.ylabel('Package(in lpa)')

#  **helps us to select a specific row or column from the data set.**

take columns starting from column 0, up to column 1, -1 means  **until the last column**.

X = df.iloc[:,0:1]

y = df.iloc[:,-1]

# splitting the data randomly training and testing

#( scikit-learn is an open-source Python library that implements a range of machine learning, pre-processing, cross-validation, and visualization algorithms using a unified interface.)

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,y\_train,y\_test = train\_test\_split(X,y,test\_size=0.2,random\_state=2)

# applying algorithm

from sklearn.linear\_model import LinearRegression

# creating object

lr = LinearRegression()

# for training model

lr.fit(X\_train,y\_train)

# converting single column data to 1 row and 1 column( the algorithm trained input shape is (1,1) one row and one column and actually we are sending 1dimension data )

lr.predict(X\_test.iloc[0].values.reshape(1,1))

# drawing best fit line

plt.scatter(df['cgpa'],df['package'])

plt.plot(X\_train,lr.predict(X\_train),color='red')

plt.xlabel('CGPA')

plt.ylabel('Package(in lpa)')

# calculating m value

m = lr.coef\_

print(m)

# calculating intercept

b = lr.intercept\_

print(b)

# y = mx + b

m \* 8.58 + b

# y = mx + b

m \* 9.5 + b