**Task 1**

**Khan Ayesha Bi**

**GRIP - The Spark Foundation**

**Data Science & Business Analytics Intern**

**Prediction using supervised learning:**

**#step1- import libraries**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

**#step2 - load data**

df = "https://bit.ly/w-data"

m = pd.read\_csv(df)

print("data imorted succesfully")

m.head()

Output:

|  | **Hours** | **Scores** |
| --- | --- | --- |
| **0** | 2.5 | 21 |
| **1** | 5.1 | 47 |
| **2** | 3.2 | 27 |
| **3** | 8.5 | 75 |
| **4** | 3.5 | 30 |

**#information of the data**

m.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 25 entries, 0 to 24

Data columns (total 2 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Hours 25 non-null float64

1 Scores 25 non-null int64

dtypes: float64(1), int64(1)

memory usage: 528.0 bytes

**#step3 plotting**

m.plot(x='Hours' , y='Scores' , style='ro')

plt.title('Hours vs percentage')

plt.xlabel('The hours studied')

plt.ylabel('The percentage Scores')

plt.show()

**#step4 dividing the data into attributes**

X = m.iloc[:, :-1].values

y = m.iloc[:, 1].values

**#step5 split of data into the training and test**

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=0)

**#step6 train the algorithm**

from sklearn.linear\_model import LinearRegression

regressor = LinearRegression()

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

print("Training ... Completed !.")

**#step7 implementing and plotting test data**

line = regressor.coef\_\*X+regressor.intercept\_

plt.scatter(X, y)

plt.plot(X, line);

plt.title('Hours vs percentage')

plt.xlabel(‘the hours studied')

plt.ylabel (‘the percentage Scores')

plt.show()

**#step8 predicting the score of model**

print(X\_test)

y\_pred = regressor.predict(X\_test)

**#step9 Comparing the actual versus predicted model to understand our model fitting**

df = pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred})

df

**#step10 time to test our model**

hours = [[9.4]]

own\_pred = regressor.predict(hours)

print("Number of hours = {}".format(hours))

print("Prediction Score = {}".format(own\_pred[0]))