**Task-2 -> To Explore Supervised Machine Learning**

In this linear regression task, have to predict the percentage of marks that a student is expected to score based upon the number of hours they studied. This is a simple linear regression task as it involves just two variables.

*# Importing all required libraries*

**from** sklearn **import** linear\_model

**import** pandas **as** pd

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

**import** numpy **as** np

**from** sklearn **import** metrics

**import** matplotlib.pyplot **as** plt

*# Reading data from remote link*

url **=** "http://bit.ly/w-data"

df **=** pd.read\_csv(url)

print("Data imported successfully")

Data imported successfully

df.head(10)

Out[3]:

|  | **Hours** | **Scores** |
| --- | --- | --- |
| **0** | 2.5 | 21 |
| **1** | 5.1 | 47 |
| **2** | 3.2 | 27 |
| **3** | 8.5 | 75 |
| **4** | 3.5 | 30 |
| **5** | 1.5 | 20 |
| **6** | 9.2 | 88 |
| **7** | 5.5 | 60 |
| **8** | 8.3 | 81 |
| **9** | 2.7 | 25 |

df

Out[4]:

|  | **Hours** | **Scores** |
| --- | --- | --- |
| **0** | 2.5 | 21 |
| **1** | 5.1 | 47 |
| **2** | 3.2 | 27 |
| **3** | 8.5 | 75 |
| **4** | 3.5 | 30 |
| **5** | 1.5 | 20 |
| **6** | 9.2 | 88 |
| **7** | 5.5 | 60 |
| **8** | 8.3 | 81 |
| **9** | 2.7 | 25 |
| **10** | 7.7 | 85 |
| **11** | 5.9 | 62 |
| **12** | 4.5 | 41 |
| **13** | 3.3 | 42 |
| **14** | 1.1 | 17 |
| **15** | 8.9 | 95 |
| **16** | 2.5 | 30 |
| **17** | 1.9 | 24 |
| **18** | 6.1 | 67 |
| **19** | 7.4 | 69 |
| **20** | 2.7 | 30 |
| **21** | 4.8 | 54 |
| **22** | 3.8 | 35 |
| **23** | 6.9 | 76 |
| **24** | 7.8 | 86 |

*# Plotting the distribution of scores*

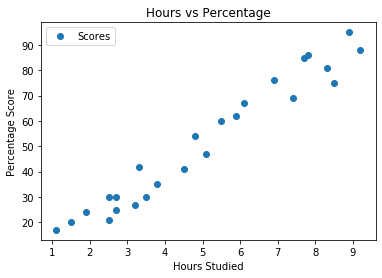
df.plot(x**=**'Hours', y**=**'Scores', style**=**'o')

plt.title('Hours vs Percentage')

plt.xlabel('Hours Studied')

plt.ylabel('Percentage Score')

plt.show()



As see there is a positive linear relation between the number of hours studied and percentage of score.

**Preparing the data**

df.iloc[5,1]

Out[6]:

20

**Splitting data into training and test**

X **=** df.iloc[:, :**-**1].values

y **=** df.iloc[:, 1].values

**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)

**Training the data**

**from** sklearn.linear\_model **import** LinearRegression

regressor **=** LinearRegression()

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

​

print("Training complete.")

Training complete.

**Plotting the regression line**

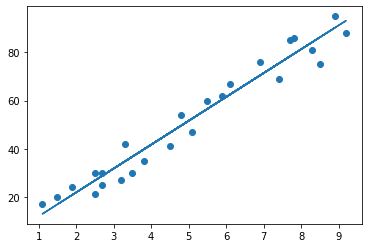
line **=** regressor.coef\_**\***X**+**regressor.intercept\_

**Plotting the test data**

plt.scatter(X, y)

plt.plot(X, line);

plt.show()



**Making predictions**

print(X\_test) *# Testing data - In Hours*

y\_pred **=** regressor.predict(X\_test) *# Predicting the scores*

[[1.5]

[3.2]

[7.4]

[2.5]

[5.9]]

**Comparing Actual vs Predicted**

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

df\_new

Out[13]:

|  | **Actual** | **Predicted** |
| --- | --- | --- |
| **0** | 20 | 16.884145 |
| **1** | 27 | 33.732261 |
| **2** | 69 | 75.357018 |
| **3** | 30 | 26.794801 |
| **4** | 62 | 60.491033 |

**Predict score if a student study for 9.25 hrs in a day**

hours **=** 9.25

own\_pred **=** regressor.predict([[hours]])

print("No of Hours = {}".format(hours))

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

No of Hours = 9.25

Predicted Score = 93.69173248737538

**Evaluating the model**

**from** sklearn **import** metrics

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Absolute Percentage Error:', np.mean(abs(y\_pred**-**y\_test)**/**y\_test))

print('Mean Square Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Square Error:',np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

Mean Absolute Error: 4.183859899002975

Mean Absolute Percentage Error: 0.12568891617045663

Mean Square Error: 21.5987693072174

Root Mean Square Error: 4.6474476121003665