The Spark Foundation Internship

Author : Deepak

Task - 1: Predict the percentage of an student based on the no. of study hours.

Prediction using Supervised Machine Learning

STEP - 1: IMPORTING REQUIRED LIBRARIES

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

STEP - 2: READING DATA

In [2]:

```
url = "http://bit.ly/w-data"
data = pd.read_csv(url)
```

In [3]:

```
data
```

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
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

In [4]:
```
data.head()
```

Out[4]:

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

In [5]:
```
#getting shape of data
data.shape
```

Out[5]:
```
(25, 2)
```

In [6]:
```
data.describe()
```

Out[6]:

	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

In [7]:
```
#getting info about data
data.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
```

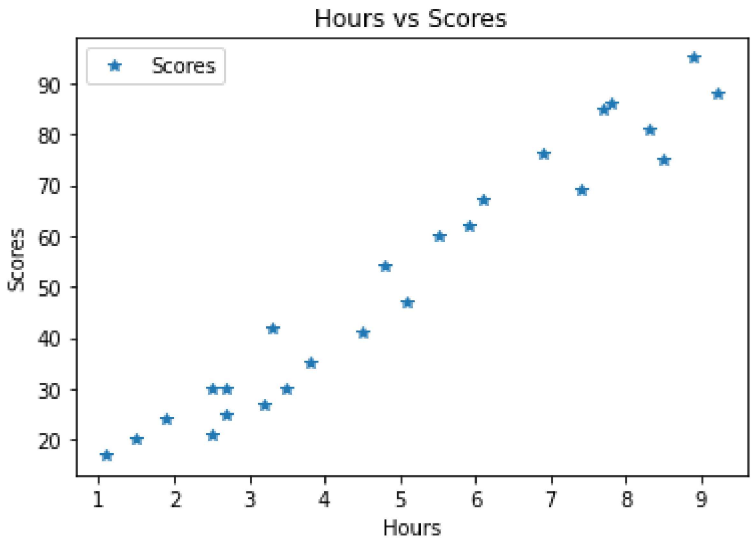
In [8]:
```
data.isnull().sum()
```

Out[8]:
```
Hours     0
Scores    0
dtype: int64
```

STEP - 3: PLOTTING THE GIVEN DATA

STEP - 3: PLOTTING THE GIVEN DATA

In [9]:
```
data.plot(x = 'Hours', y = 'Scores', style = '*')
plt.title("Hours vs Scores")
plt.xlabel("Hours")
plt.ylabel("Scores")
plt.show()
```



In [10]:
```
# using iloc function we will divide the data
X = data.iloc[:, :-1].values  #for hours
y = data.iloc[:, 1].values    #for scores
```

STEP - 4: SPLITTING THE DATA FOR TRAINING AND TESTING

In [11]:
```
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_sta
```

STEP - 5: TRAINING THE MODEL

In [12]:
```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x_train, y_train)
```

Out[12]:
```
LinearRegression()
```

STEP - 6: PREDICTING THE TEST SCORES

In [13]:
```
y_pred = model.predict(x_test)
```

In [14]:
```
y_pred
```

Out[14]:
```
array([16.88414476, 33.73226078, 75.357018  , 26.79480124, 60.49103328])
```

In [15]:

```
df = pd.DataFrame({"Actual Score":y_test,"Predicted Score":y_pred})
df
```

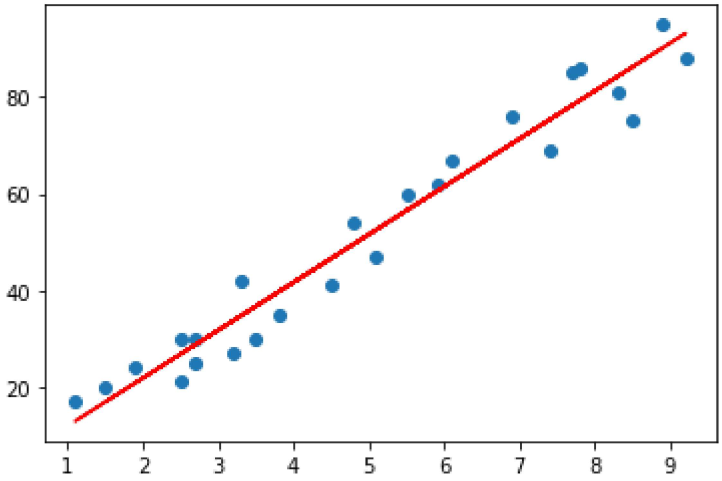
Out[15]:

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

STEP 7 - VISUALIZING THE MODEL

In [16]:

```
line = model.coef_*X + model.intercept_
plt.scatter(X, y)
plt.plot(X, line,color = "r")
plt.show()
```



STEP - 8: PREDICTING THE VALUE FOR THE GIVEN HOURS

In [17]:

```
hours = 9.25
res = model.predict([[hours]])
print(f"The number of hours is {hours}")
print(f"The predicted value is {res[0]}")
```

```
The number of hours is 9.25
The predicted value is 93.69173248737538
```

STEP - 9: EVALUATING THE MODEL

In [18]:
```
from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
```

```
Mean Absolute Error: 4.183859899002975
```

THANK YOU