Name: Romita Bandyopadhyay

Data Science and Business Analytics Intern under The Sparks Foundation

Task 1: Prediction using Supervised ML

Prediction of percentage of a student based on number of hours studied.

Step 1: Importing Required Libraries and Given Data Set

In [1]:

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

In [2]:

```
df=pd.read_csv('http://bit.ly/w-data')
df.head() # Showing first 5 values
```

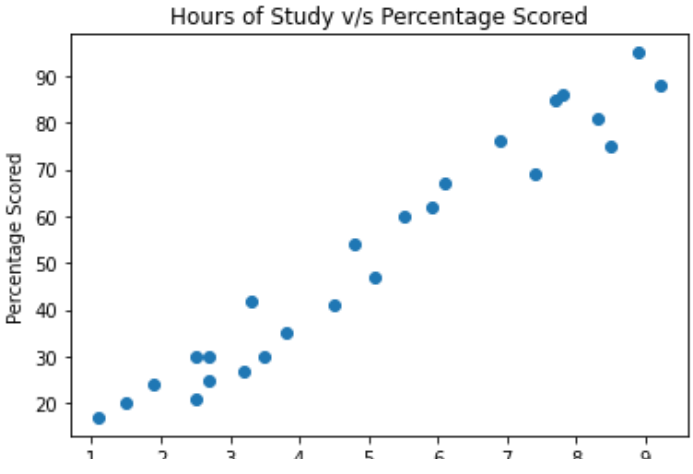
Out[2]:

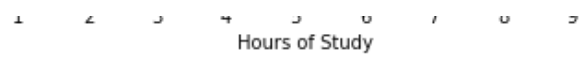
	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

Step 2: Visualization of Dataset

In [3]:

```
plt.scatter(df['Hours'], df['Scores'])
plt.xlabel('Hours of Study')
plt.ylabel('Percentage Scored')
plt.title('Hours of Study v/s Percentage Scored')
plt.show()
```





Correlation among variables is calculated to check the existence of any linear relationship

In [4]:

```
df.corr()
```

Out[4]:

	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

The value 0.976 makes it evident that there exists a linear relationship between scores and hours spend on studying.

Step 3: Spliting Dataset Values

In [5]:

```
X=df[['Hours']] # independent variables
Y=df['Scores']  # dependent variables
```

In [6]:

```
X.head() # Showing first 5 values of independent variable.
```

Out[6]:

	Hours
0	2.5
1	5.1
2	3.2
3	8.5
4	3.5

In [7]:

```
Y.head() #Showing first 5 values of dependent variable.
```

Out[7]:

```
0    21
1    47
2    27
3    75
4    30
Name: Scores, dtype: int64
```

In [8]:

```
from sklearn.model_selection import train_test_split
```

In [9]:

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, random_state
= 0)
# The training and test data is splitted in 80:20 ratio
```

Training of the Model

LInear Regression Model is imported

In [10]:

```
from sklearn.linear_model import LinearRegression
```

An object is created for our model.

In [11]:

```
reg = LinearRegression()
```

In [12]:

```
reg.fit(X_train, Y_train)
```

Out[12]:

```
LinearRegression()
```

Step 5: Testing model on test Dataset

In [13]:

```
X_test
```

Out[13]:

	Hours
5	**1.5**
2	**3.2**
19	**7.4**
16	**2.5**
11	**5.9**

In [14]:

```
Y_test
```

Out[14]:

```
5     20
2     27
19    69
16    30
11    62
Name: Scores, dtype: int64
```

In [15]:

```
pred= reg.predict(X_test)
pred
```

Out[15]:

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

Regression coeffecient is calculated to check the accuracy of our model.

In [16]:

```
reg.score(X_test, Y_test)
```

Out[16]:

```
0.9454906892105355
```

The model is 94% accurate.

Step 6: Visualization of model with coefficient and intercept values.

In [17]:

```
reg.coef_
```

Out[17]:

```
array([9.91065648])
```

In [18]:

```
reg.intercept_
```

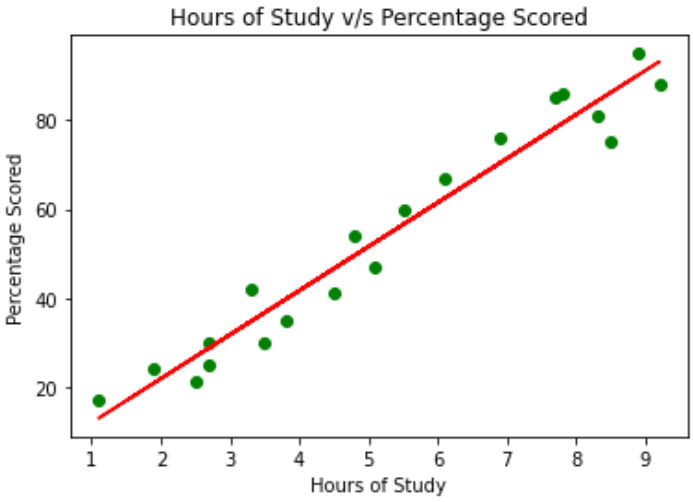
Out[18]:

```
2.018160041434669
```

In [19]:

```
# Visualizing the model

lrline = reg.coef_*X + reg.intercept_
plt.show()
plt.scatter(X_train, Y_train, color = 'green')
plt.plot(X, lrline, color='red')
plt.xlabel('Hours of Study')
plt.ylabel('Percentage Scored')
plt.title('Hours of Study v/s Percentage Scored')
plt.show()
```



Step 7: Prediction of percentage.

In [20]:

```
hours = 9.25
studyinghours=[[hours]]
percentage = reg.predict((studyinghours))
print('Predicted Percentage='+ str(percentage) + '%')
```

```
Predicted Percentage=[93.69173249]%
```

In [21]:

```
print ('A student studying for '+ str(hours) + ' hours per day is expected to score a per
centage of '+ str(percentage)+ '%, approximately')
```

```
A student studying for 9.25 hours per day is expected to score a percentage of [93.691732
49]%, approximately
```

Step 8: Evaluating Model.

In [22]:

```
from sklearn import metrics
print ('Mean Absolute Error = ' + str(metrics.mean_absolute_error(Y_test, pred)))
```

```
Mean Absolute Error =4.18385989900298
```

Task 1 completes here.

In [ ]:

```
```