The Spark Foundation - Data Science and Business Analytics

TASK 1 - Prediction using Supervised ML

Name-Asmita Deb

Predict the percentage of an student based on the no. of study hours.This is a simple linear regression task as it involves just 2 variables. What will be predicted score if a student studies for 9.25Hrs/day?

In [1]:
```
#STEP 1 - importing the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [6]:
```
#STEP 2 - importing the data
link ="https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_scores%20-%20student_scores.csv"
df = pd.read_csv(link)
df
```

Out[6]:

	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 [7]:
```
df.shape
```

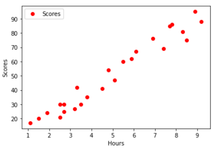
Out[7]:
```
(25, 2)
```

In [8]:
```
#STEP 3 - Checking if the data is clean or not
df.isnull().sum()
```

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

In [9]:
```
#STEP 4 - Plotting the data visualization
df.plot(x='Hours',y='Scores',style = 'ro')
plt.xlabel('Hours')
plt.ylabel('Scores')
plt.show
```

Out[9]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



we can clearly see that there is a positive linear relation between the number of hours studied and percentage of score.

In [ ]:
```
#STEP 5 - PREPARING LINEAR REGRESSION MODEL
```

In [31]:
```
X = df.iloc[:, :-1].values
y = df.iloc[:, 1].values
```

In [32]:
```
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)
```

In [51]:
```
len(X_train)
```

Out[51]:
```
20
```

In [52]:
```
len(X_test)
```

Out[52]:
```
5
```

In [53]:
```
len(y_train)
```

Out[53]:
```
20
```

In [54]:
```
len(y_test)
```

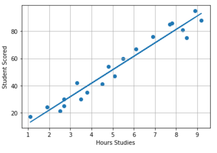
Out[54]:
```
5
```

In [34]:
```
#STEP 6 - Training the model
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
```

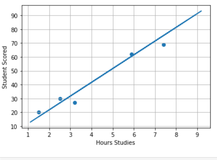
In [35]:
```
lr.fit( X_train , y_train)
```

Out[35]:
```
LinearRegression()
```

In [26]:
```
#STEP 7 - PLOTTING THE REGRESSION LINE
line = lr.coef_*X+lr.intercept_
#plotting the training data
plt.scatter(X_train, y_train)
plt.plot(X, line);
plt.xlabel('Hours Studies')
plt.ylabel('Student Scored')
plt.grid()
plt.show()
```



In [25]:
```
line = lr.coef_*X+lr.intercept_
#plotting the test data
plt.scatter(X_test, y_test)
plt.plot(X, line);
plt.xlabel('Hours Studies')
plt.ylabel('Student Scored')
plt.grid()
plt.show()
```



In [35]:
```
#STEP 8 - MAKING THE PREDICTIONS

print(X_test) #testing data in hours
y_predict = lr.predict(X_test) #predicting the scores
```

```
[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]]
```

In [36]:
```
y_test
```

Out[36]:
```
array([20, 27, 69, 30, 62], dtype=int64)
```

In [37]:
```
y_predict
```

Out[37]:
```
array([16.88414476, 33.73226078, 75.357018  , 26.79480124, 60.49103328])
```

In [38]:
```
data= pd.DataFrame({'Actual': y_test,'Predicted': y_predict})
data
```

Out[38]:

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

In [40]:
```
#LETS PREDICT FOR 9.25 HRS
print('Score of student who studied for 9.25 hours a dat', lr.predict([[9.25]]))
```

```
Score of student who studied for 9.25 hours a dat [93.69173249]
```

In [44]:
```
#STEP - 9 MODEL EVALUATION
from sklearn import metrics
print ('Mean Absolute Error-', metrics.mean_absolute_error(y_test, y_predict))
```

```
Mean Absolute Error- 4.183859899002975
```

In [ ]:
```
```