Mayur Shinde

Data Science and Business Analytics Intern @ The Sparks Foundation

Topic : Prediction using Supervised ML

Dataset : http://bit.ly/w-data

In [1]:

```
# GRIP Task 1 by Mayur Shinde
# Prediction using Supervised ML
```

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
```

In [3]:

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

In [5]:

```
# to check if any null data is present or not
data.isnull == True
```

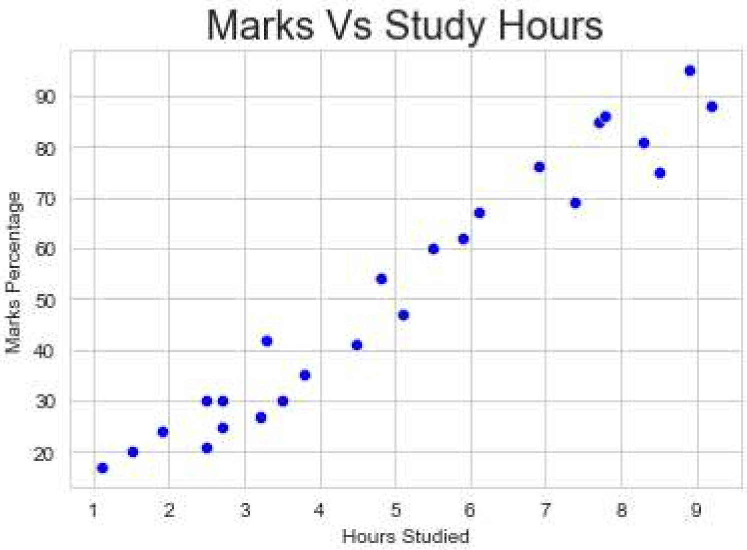
Out[5]:

```
False
```

In [43]:

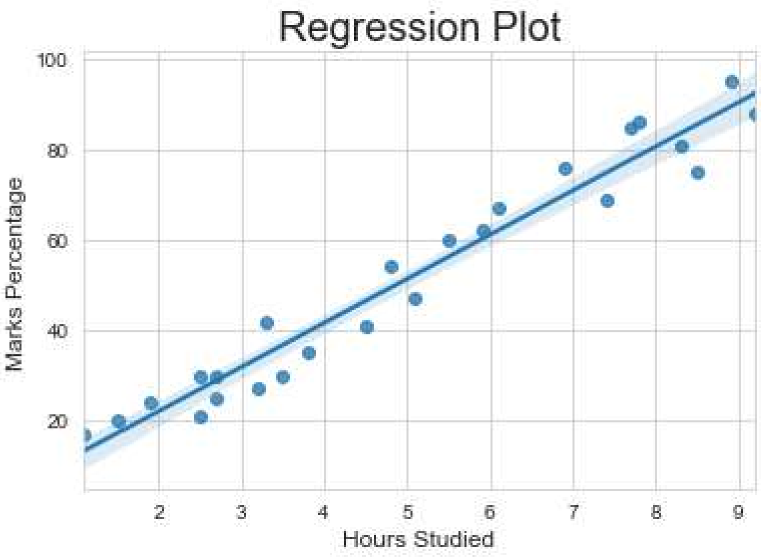
```
sns.set_style('whitegrid')
sns.scatterplot(y= data['Scores'], x= data['Hours'], color='Blue')
```

```
plt.title('Marks Vs Study Hours',size=20)
plt.ylabel('Marks Percentage', size=10)
plt.xlabel('Hours Studied', size=10)
plt.show()
```



In [36]:

```
sns.regplot(x= data['Hours'], y= data['Scores'])
plt.title('Regression Plot',size=20)
plt.ylabel('Marks Percentage', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
print(data.corr())
```



```
           Hours    Scores
Hours   1.000000  0.976191
Scores  0.976191  1.000000
```

Training the Model

1] Splitting the Data

In [20]:

```
# Defining X and y from the Data
X = data.iloc[:, :-1].values
```

```
y = data.iloc[:, 1].values

# Spliting the Data in two
train_X, val_X, train_y, val_y = train_test_split(X, y, random_state = 0)
```

2] Fitting the Data into the Model

In [21]:

```
regression = LinearRegression()
regression.fit(train_X, train_y)
print("---------Model Trained---------")
```

```
---------Model Trained---------
```

Predicting the Percentage of Marks

In [22]:

```
pred_y = regression.predict(val_X)
prediction = pd.DataFrame({'Hours': [i[0] for i in val_X], 'Predicted Marks': [k for k
prediction
```

Out[22]:

	Hours	Predicted Marks
0	1.5	16.844722
1	3.2	33.745575
2	7.4	75.500624
3	2.5	26.786400
4	5.9	60.588106
5	3.8	39.710582
6	1.9	20.821393

Comparing the Predicted Marks with the Actual Marks

In [23]:

```
compare_scores = pd.DataFrame({'Actual Marks': val_y, 'Predicted Marks': pred_y})
compare_scores
```

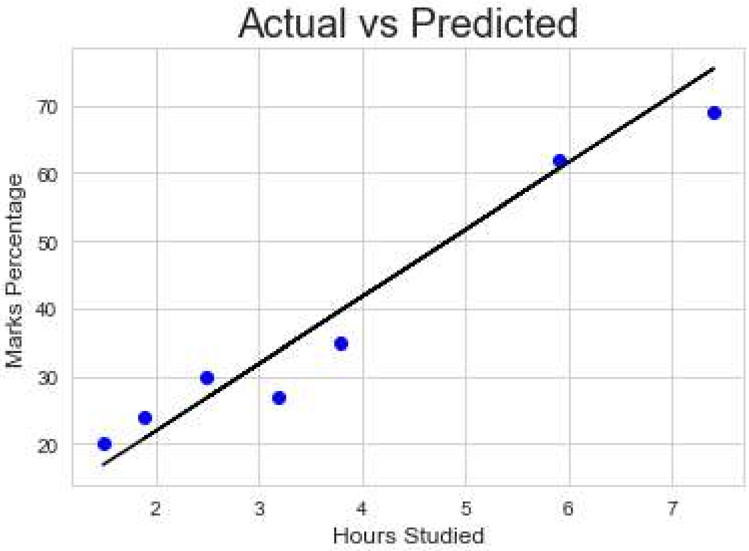
Out[23]:

	Actual Marks	Predicted Marks
0	20	16.844722
1	27	33.745575
2	69	75.500624
3	30	26.786400
4	62	60.588106
5	35	39.710582
6	24	20.821393

Visually Comparing the Predicted Marks with the Actual Marks

In [32]:

```
plt.scatter(x=val_X, y=val_y, color='Blue')
plt.plot(val_X, pred_y, color='Black')
plt.title('Actual vs Predicted', size=20)
plt.ylabel('Marks Percentage', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
```



Evaluating the Model

In [37]:

```
# Calculating the accuracy of the model
print('Mean absolute error: ',mean_absolute_error(val_y,pred_y))
```

```
Mean absolute error:  4.130879918502486
```

What will be the predicted score of a student if he/she studies for 9.25 hrs/ day?

In [38]:

```
hours = [9.25]
answer = regression.predict([hours])
print("Score = {}".format(round(answer[0],3)))
```

```
Score = 93.893
```

According to the regression model if a student studies for 9.25 hours a day he/she is likely to score 93.89 marks.