Y.Vaishnavi

Task 1 -Prediction using Supervised Machine learning

In [21]:

```
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 [23]:

```
url="http://bit.ly/w-data"
sd=pd.read_csv(url)#Reading data from data set given
print("DATA IMPORTED")
sd.head(5)
```

```
DATA IMPORTED
```

Out[23]:

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

In [25]:

```
sd.tail() #prints last 5 data in dataset
```

Out[25]:

	Hours	Scores
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

In [26]:

```
sd.isnull==True
```

Out[26]:

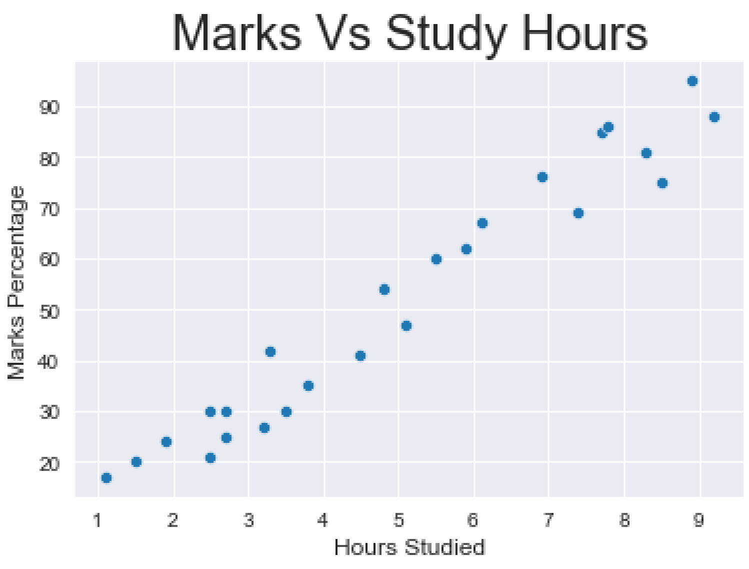
```
False
```

In [27]:

```
sns.set_style('darkgrid')
```

In [53]:

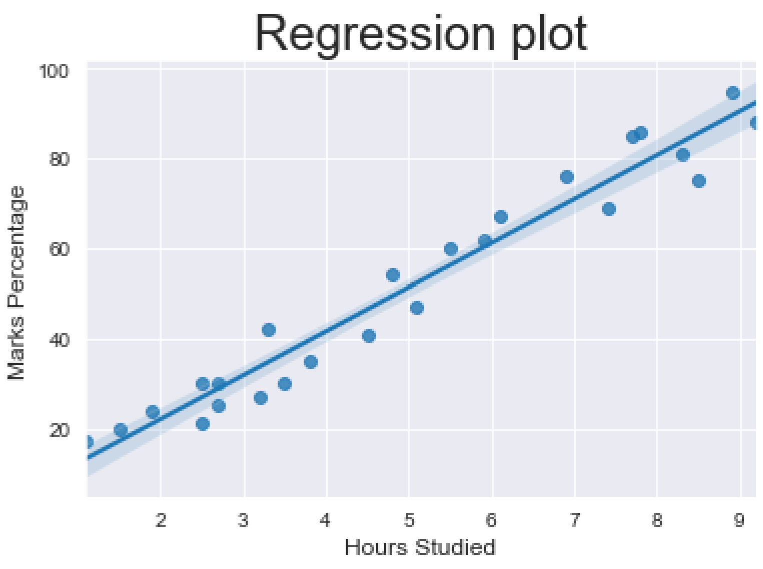
```
sns.scatterplot(y= sd['Scores'], x= sd['Hours'])
plt.title('Marks Vs Study Hours',size=24)
plt.ylabel('Marks Percentage', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
```



From the above scatter plot there looks to be correlation between the 'Marks Percentage' and 'Hours Studied',Lets plot a regression line to confirm correlation.

In [64]:

```
sns.regplot(x= sd['Hours'] , y= sd['Scores'])
plt.title('Regression plot',size=24)
plt.ylabel('Marks Percentage', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
print(sd.corr())
print("THE VARIABLES ARE POSITIVELY CORRELATED")
```



```
           Hours    Scores
Hours   1.000000  0.976191
Scores  0.976191  1.000000
THE VARIABLES ARE POSITIVELY CORRELATED
```

Traning Model

In [66]:

```
#defining x and y from the data
X=sd.iloc[:, :-1].values
y=sd.iloc[:, 1].values

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

In [67]:

```
##Fitting the data into the model
regression = LinearRegression()
regression.fit(train_X, train_y)
```

Out[67]:

```
LinearRegression()
```

PREDICTING THE PERCENTAGE OF MARKS

In [72]:

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

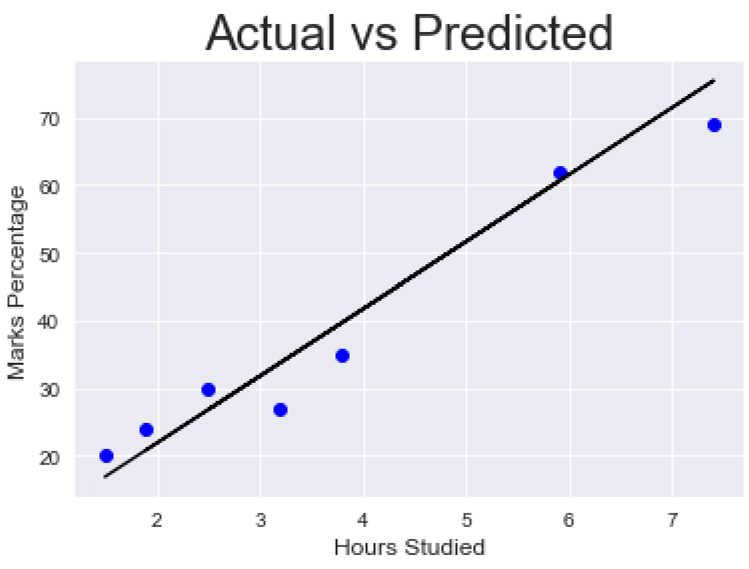
Out[72]:

	Hours	Predicting 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

Visually comparing the predicting marks with the Actual Marks

In [75]:

```
plt.scatter(x=val_X, y=val_y, color='blue')
plt.plot(val_X, predicting_y, color='Black')
plt.title('Actual vs Predicted', size=24)
plt.ylabel('Marks Percentage', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
```



Evaluating the Model

In [79]:

```
# Calculating the accuracy of the model
print('Mean absolute error: ',mean_absolute_error(val_y
,predicting_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 [77]:

```
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.