TASK 1 - Prediction using supervised Machine Learning¶

Done by siddharth

In [53]:

```
#importing the required libraries
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 [54]:

```
#reading data
db = pd.read_csv('http://bit.ly/w-data')
db.tail(10)
```

Out[54]:

	Hours	Scores
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 [55]:

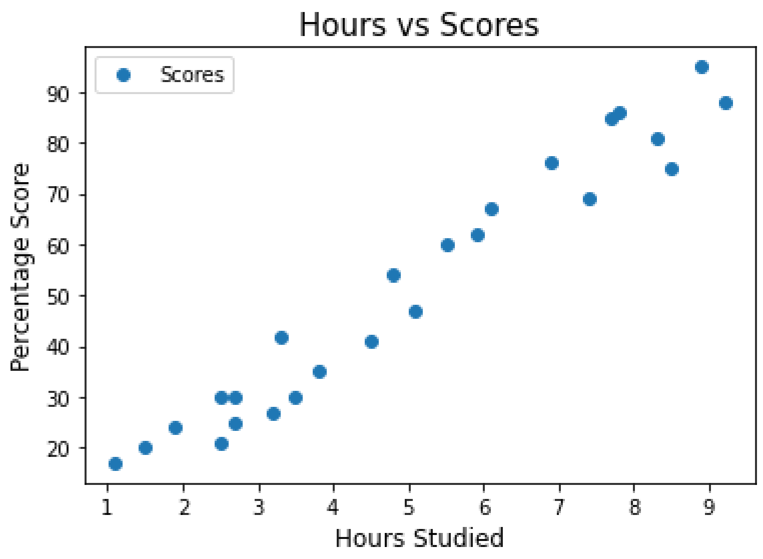
```
#checking null value in dataset
db.isnull == True
```

Out[55]:

```
False
```

In [56]:

```
#visualizing the data
db.plot(x = 'Hours', y = 'Scores', style = 'o')
plt.title('Hours vs Scores', size = 15)
plt.xlabel('Hours Studied', size = 12)
plt.ylabel('Percentage Score', size = 12)
plt.show()
```



In [57]:

```
#dividing the data into 'attributes' and 'labels'
x = db.iloc[:, :-1].values
y = db.iloc[:, 1].values

#spliting the data into training and test sets.
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2, random_state = 0)
```

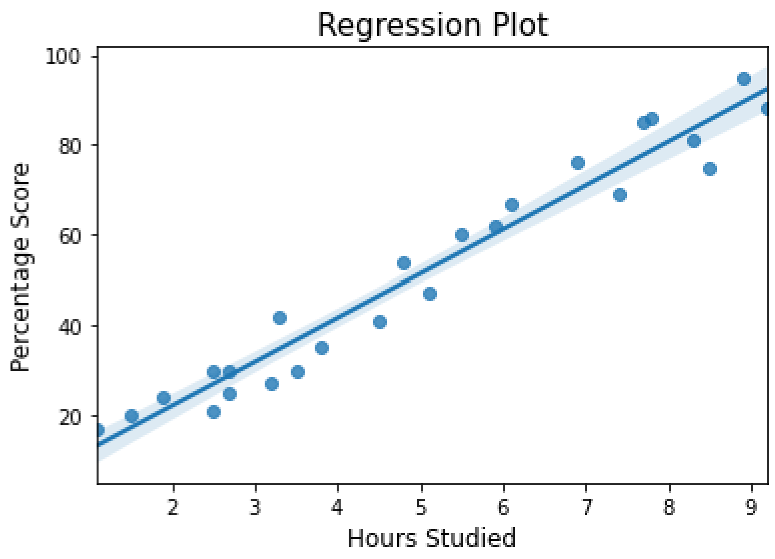
In [58]:

```
#training the model
regression = LinearRegression()
regression.fit(x_train, y_train)

print("model trained...")
```

```
model trained...
```

In [59]:

```
sns.regplot(x = db['Hours'], y = db['Scores'])
plt.title('Regression Plot', size = 15)
plt.xlabel('Hours Studied', size = 12)
plt.ylabel('Percentage Score', size = 12)
plt.show()
```



In [60]:

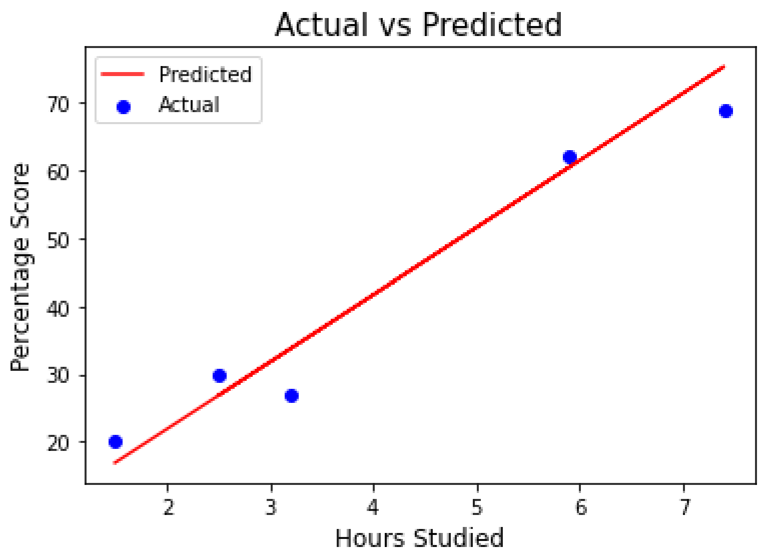
```
#now predicting the percentage score
y_pred = regression.predict(x_test)

#comparing the predicted score with actual score
prediction = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
prediction
```

Out[60]:

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

In [61]:

```
#visualizing predicted score vs actual score
plt.scatter(x = x_test, y = y_test, color = 'blue', label = 'Actual')
plt.plot(x_test, y_pred, color = 'red', label = 'Predicted')
plt.legend()
plt.title('Actual vs Predicted', size=15)
plt.ylabel('Percentage Score', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
```



In [62]:

```
#evaluating the model, calculating the accuracy of the model
print('Mean absolute error: ', mean_absolute_error(y_test, y_pred))
```

```
Mean absolute error:  4.183859899002975
```

In [67]:

```
#What will be predicted score if a student studies for 9.25 hrs/ day?
hours = [8.3]
ans = regression.predict([hours])
print("Score = {}".format(round(ans[0],3)))
```

```
Score = 84.277
```

In [ ]: