Name: Yassen Ayman // Track: Data Science and Bussiness Analytics

predection using supervised ML for Sparks intern

In [43]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
data=pd.read_csv('hoursScore.txt')
```

In [4]:
```
data.head()
```

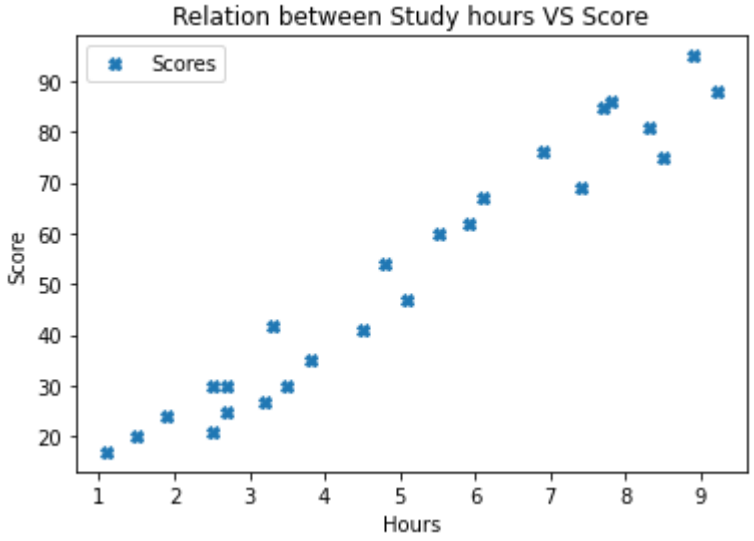
Out[4]:

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

In [5]:
```
data.info()
```

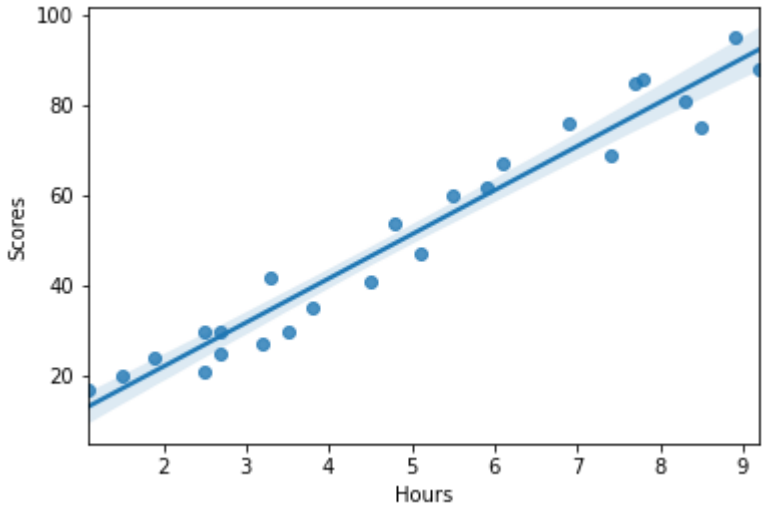
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Hours   25 non-null     float64
 1   Scores  25 non-null     int64
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

In [34]:
```
data.plot(x='Hours',y='Scores',style='X')
plt.xlabel('Hours')
plt.ylabel('Score')
plt.title('Relation between Study hours VS Score')
plt.show()
```



In [45]:
```
sns.regplot(data=data,x='Hours',y='Scores')
```

Out[45]:
```
<AxesSubplot:xlabel='Hours', ylabel='Scores'>
```



Scaling

In [32]:
```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
newdata=scaler.fit_transform(data)
# convert Data from array to dataframe
newdata=pd.DataFrame(newdata)
```

split

In [82]:
```
from sklearn.model_selection import train_test_split
x=data.iloc[:,:-1]
y=data.iloc[:,1]
xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.1,random_state=44,shuf
```

Model

In [83]:
```
from sklearn.linear_model import LinearRegression
LR=LinearRegression()
# Fitting Simple Linear Regression to the Training set
LR.fit(xtrain,ytrain)
LR.score(xtrain,ytrain)
LR.score(xtest,ytest)
```

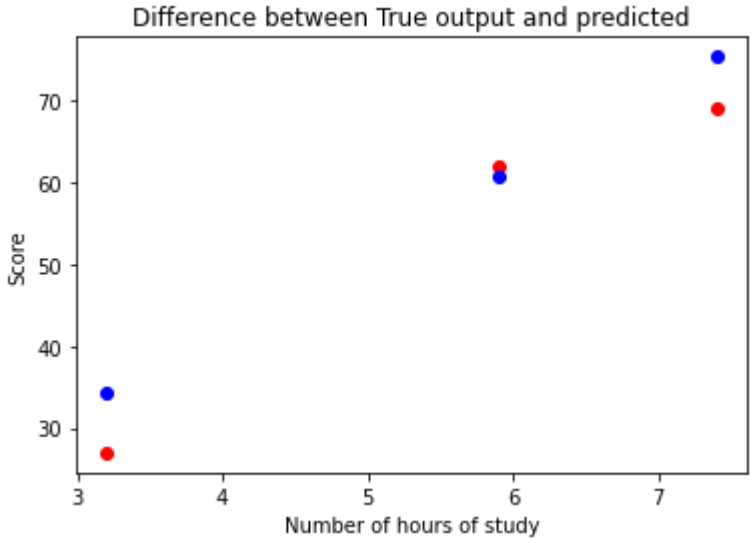
Out[83]:
```
0.9068477270479203
```

In [84]:
```
y_predicted=LR.predict(xtest)
y_predicted
```

Out[84]:
```
array([75.32587159, 34.24604687, 60.65450562])
```

Difference between True output and predicted

In [85]:
```
plt.scatter(xtest['Hours'],ytest,c='r')
plt.scatter(xtest['Hours'],y_predicted,c='b')
plt.xlabel('Number of hours of study')
plt.ylabel('Score')
plt.title('Difference between True output and predicted')
plt.show()
```



Test with specific value

In [89]:
```
hours=[[9.25]]
predic_value=LR.predict(hours)
print("Score based on hours is :",predic_value[0])
```

```
Score based on hours is : 93.42055629643744
```

Evaluting The Model

In [90]:
```
from sklearn.metrics import  mean_absolute_error
print('Mean Absolute Error is:', mean_absolute_error(ytest, y_predicted))
```

```
Mean Absolute Error is: 4.972470946847014
```

In [ ]: