# Data Science and Business Analytics Intern @The Sparks Foundation

# NAME - ATANU DAS

# TASK 1 : Prediction Using Supervised ML(Level - Beginner)

# Predict the percentage of an student based on the no. of study hours. What will be predicted score if a student studies for 9.25hrs/day?

## Importing Libraries

In [21]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
```

## Reading Data

In [24]:
```
url = "http://bit.ly/w-data"
data=pd.read_csv(url)
```

In [25]:
```
#Printing the shape of a dataset
data.shape
```

Out[25]: (25, 2)

In [6]:
```
#Statistical summary of the data
data.describe()
```

Out[6]:

| | Hours | Scores |
|---|---|---|
| **count** | 25.000000 | 25.000000 |
| **mean** | 5.012000 | 51.480000 |
| **std** | 2.525094 | 25.286887 |
| **min** | 1.100000 | 17.000000 |

| | Hours | Scores |
|---|---|---|
| **25%** | 2.700000 | 30.000000 |
| **50%** | 4.800000 | 47.000000 |
| **75%** | 7.400000 | 75.000000 |
| **max** | 9.200000 | 95.000000 |

In [7]:
```
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
```

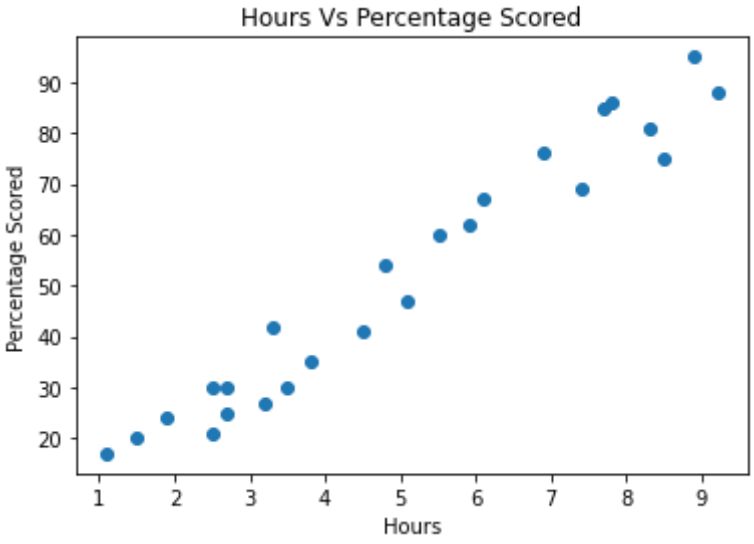
In [8]:
```
#Checking whether the data is having null value or not
data.isnull().sum()
```

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

# Data Visualization

In [9]:
```
x="Hours"
y="Scores"
plt.scatter(x,y,data=data)
plt.title("Hours Vs Percentage Scored")
plt.xlabel("Hours")
plt.ylabel("Percentage Scored")
```

Out[9]:
```
Text(0, 0.5, 'Percentage Scored')
```




In [10]:
```
#Reshaping the Hours and Scores column in to array
X=data.iloc[:, :-1].values
```

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

In [11]:
```
#Train and Test Data
X_train,X_test,y_train,y_test=train_test_split(X,Y,test_size=0.2,random_state=0)
```

In [12]:
```
X_train.shape
y_train.shape
```

Out[12]: (20,)

In [13]:
```
X_test.shape
y_test.shape
```

Out[13]: (5,)

# Linear Regression

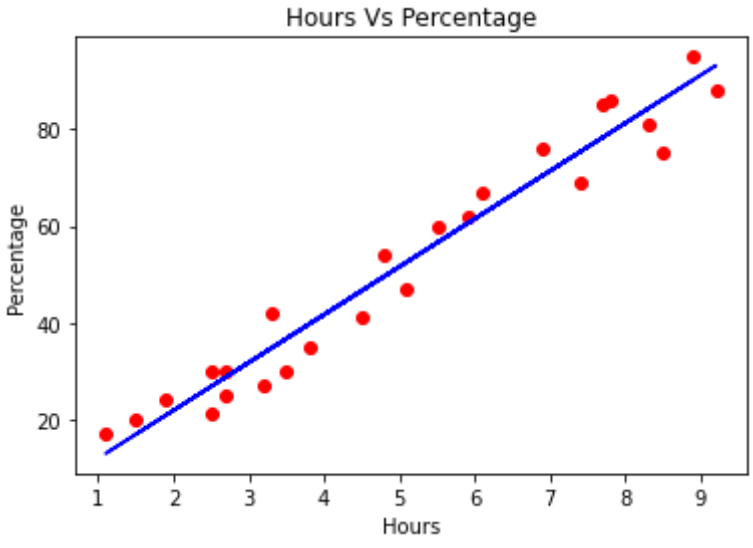
In [14]:
```
reg=LinearRegression()
print(reg)
```

```
LinearRegression()
```

In [15]:
```
reg.fit(X_train,y_train)
```

Out[15]:
```
LinearRegression()
```

In [16]:
```
#Plotting the regression line
l=reg.coef_*X + reg.intercept_
plt.scatter(X,Y,color='r')
plt.plot(X,l,color='b')
plt.title("Hours Vs Percentage")
plt.xlabel("Hours")
plt.ylabel("Percentage")
plt.show()
```




In [17]:
```
print(X_test)
```

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

In [18]:
```
y_prediction=reg.predict(X_test)
d=pd.DataFrame({'Actual':y_test,'Predicted':y_prediction})
d
```

Out[18]:

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

# Predicted Score if a student studies for 9.25hrs/day

In [19]:
```
ans=reg.predict([[9.25]])
print("The predicted score is {} if a student studies for 9.25hrs/day".format(ans[0]
```

```
The predicted score is 93.69173248737538 if a student studies for 9.25hrs/day
```

# Accuracy Check

In [20]:
```
print("Mean Absolute Error : ",mean_absolute_error(y_test,y_prediction))
```

```
Mean Absolute Error :  4.183859899002975
```

In [ ]: