In [1]:

```
### Arun kumar K

### The Sparks Foundation

### GRIP - Graduate Rotational Internship Program

### Task - 1

### Prediction using Supervised Learning

# Importing the libraries



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



# Read the file

url="http://bit.ly/w-data"
df = pd.read_csv(url)
print(df)
```

```
Matplotlib is building the font cache; this may take a moment.
    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
10    7.7      85
11    5.9      62
12    4.5      41
13    3.3      42
14    1.1      17
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 [3]:

```
df.shape
```

Out[3]:

```
(25, 2)
```

In [4]:

```
df.describe()
```

Out[4]:

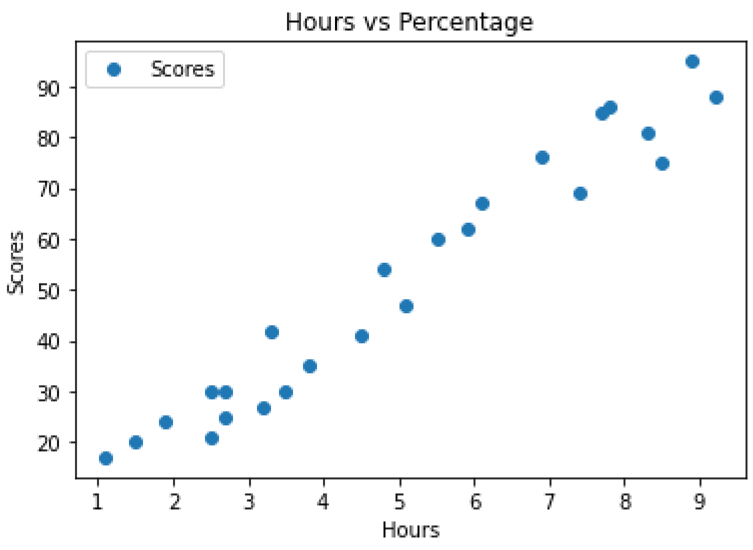
	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

In [ ]:

```
# Ploting the dataset
```

In [5]:

```
df.plot(x='Hours', y='Scores',style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours')
plt.ylabel('Scores')
plt.show()
```



In [ ]:

```
# Test and Train Dataset
```

In [8]:

```
X = df.iloc[:,:-1].values
y = df.iloc[:,1].values
```

In [9]:

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=0)
```

In [10]:

```
regressor = LinearRegression()
regressor.fit(X_train,y_train)
```

Out[10]:

```
LinearRegression()
```

In [11]:

```
regressor.coef_
```

Out[11]:

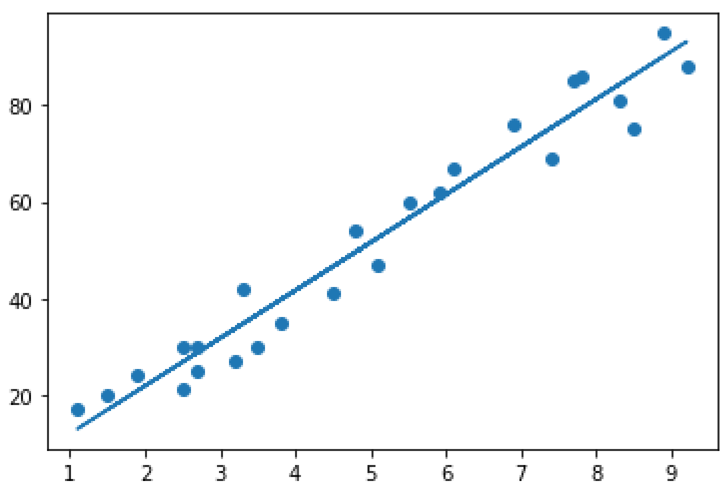
```
array([9.91065648])
```

In [12]:

```
# Scatter Plot for the test data using the trained data
```

In [13]:

```
line = regressor.coef_*X+regressor.intercept_
plt.scatter(X,y)
plt.plot(X,line);
plt.show()
```



In [ ]:

```
# Prediction of the scores
```

In [14]:

```
print(X_test)
y_pred = regressor.predict(X_test)
```

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

In [ ]:

```
# Comparing the models (Actual vs Predicted)
```

In [15]:

```
dataset=pd.DataFrame({'Actual': y_test,'Predicted': y_pred})
dataset
```

Out[15]:

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

In [ ]:

```
# Predicting the conditions (Hours = 9.25 per day)
```

In [16]:

```
Hours=[[9.25]]
own_pred=regressor.predict(Hours)
print("Number of Hours ={}".format(Hours))
print("Prediction Score ={}".format(own_pred[0]))
```

```
Number of Hours =[[9.25]]
Prediction Score =93.69173248737538
```

In [ ]:

```
# Mean Absolute Error
```

In [17]:

```
from sklearn import metrics
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test,y_pred))
```

```
Mean Absolute Error: 4.183859899002975
```