TASK1 : Prediction using Supervised ML

AUTHOR : KEERTHANA

Importing Libraries and Packages

In [21]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
```

Import Dataset

In [22]:

```
url="https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_scores%2
dataset1 = pd.read_csv(url)
dataset1
```

Out[22]:

	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

Check for NULL values

In [23]:

```
dataset1.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
```

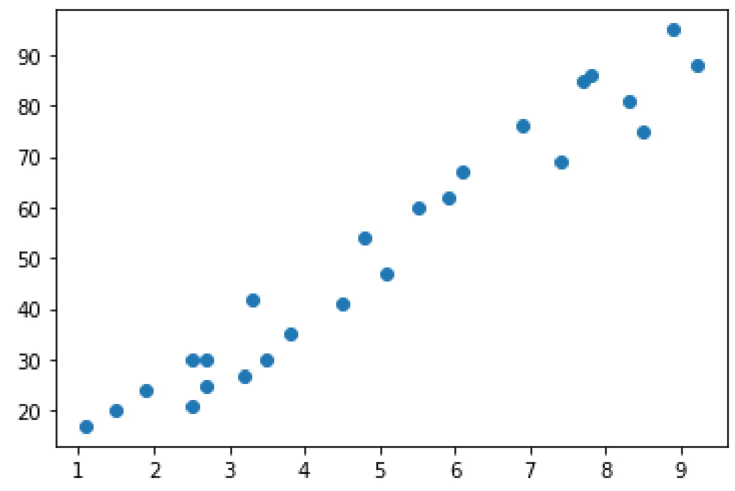
Visualizing the data

In [24]:

```
x=dataset1['Hours']
y=dataset1['Scores']
plt.scatter(x,y)
```

Out[24]:

```
<matplotlib.collections.PathCollection at 0x16c20fc8610>
```



Performing Linear Regression

Training and Testing Sets

In [25]:

```
x=dataset1['Hours']
y=dataset1['Scores']
```

In [26]:

```
x_train, x_test, y_train, y_test = train_test_split(x, y, train_size = 0.5,test_size =
```

In [27]:

```
x_train
```

Out[27]:

```
23    6.9
13    3.3
19    7.4
1     5.1
17    1.9
7     5.5
4     3.5
2     3.2
14    1.1
6     9.2
9     2.7
24    7.8
Name: Hours, dtype: float64
```

In [28]:

```
y_train
```

Out[28]:

```
23    76
13    42
19    69
1     47
17    24
7     60
4     30
2     27
14    17
6     88
9     25
24    86
Name: Scores, dtype: int64
```

In [29]:

```
x_train = x_train.values.reshape(-1,1)
x_test = x_test.values.reshape(-1,1)
print(x_train.shape)
print(x_test.shape)
```

```
(12, 1)
(13, 1)
```

In [30]:

```
lm = LinearRegression()
lm.fit(x_train, y_train)
```

Out[30]:

```
LinearRegression()
```

Slope and Intercept

In [31]:

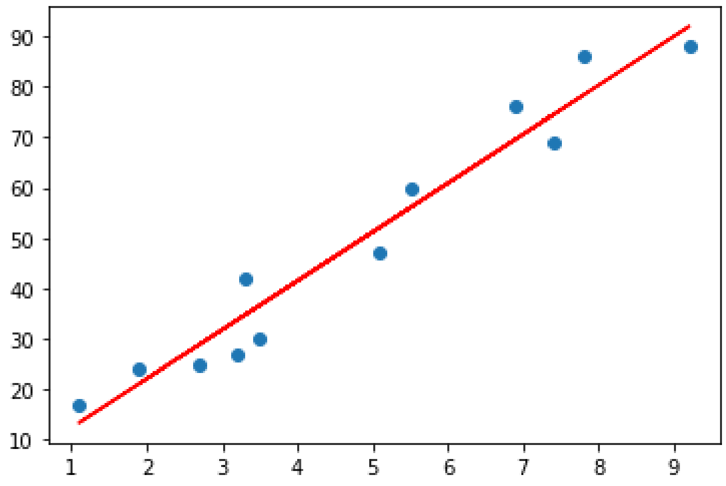
```
print("Intercept :",lm.intercept_)
print('Slope :',lm.coef_)
```

```
Intercept : 2.6351203501093963
Slope : [9.71143326]
```

Plotting the graph

In [32]:

```
res=lm.intercept_+lm.coef_*x_train
plt.scatter(x_train, y_train)
plt.plot(x_train, res, 'r')
plt.show()
```



Predicted Response

In [33]:

```
print('predicted response:',res, sep='\n')
```

```
predicted response:
[[69.64400985]
 [34.68285011]
 [74.49972648]
 [52.16342998]
 [21.08684354]
 [56.04800328]
 [36.62513676]
 [33.71170678]
 [13.31769694]
 [91.98030635]
 [28.85599015]
 [78.38429978]]
```

In [ ]:

```

```