LINEAR REGRESSION

TASK-1:Predict the percentage of marks of an student based on the number of study hours

NAME: JAYAKUMAR

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
data=pd.read_csv('http://bit.ly/w-data')
```

In [4]:
```
data
```

Out[4]:

	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 [5]:
```
data.head()
```

Out[5]:

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

In [6]:
```
data.tail()
```

Out[6]:

	Hours	Scores
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

In [7]:
```
data.describe()
```

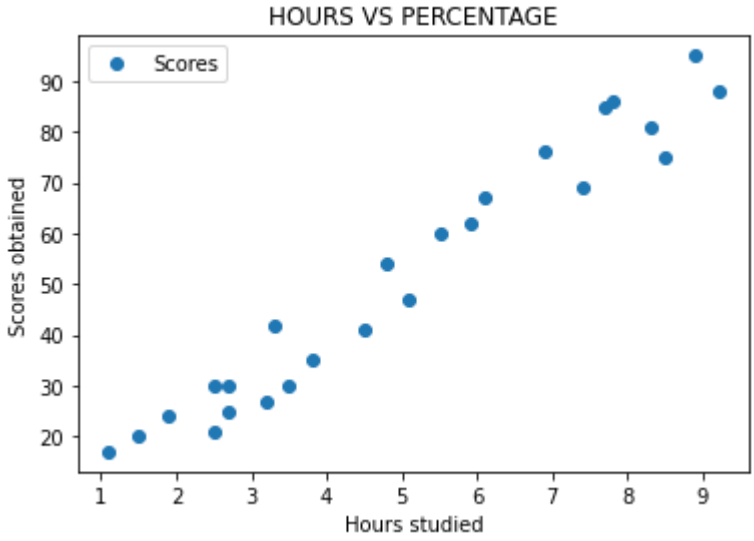
Out[7]:

	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 [8]:
```
data.shape
```

Out[8]:
```
(25, 2)
```

In [10]:
```
data.plot(x='Hours',y='Scores',style='o')
plt.title("HOURS VS PERCENTAGE")
plt.xlabel("Hours studied")
plt.ylabel('Scores obtained')
plt.show()
```



Training the algorithm

In [11]:
```
X=data.iloc[:,:-1].values
Y=data.iloc[:,1].values
```

Splitting data into train and test sets for training

In [12]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(X,Y,test_size=0.2,random_state=
0)
```

In [19]:
```
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
```

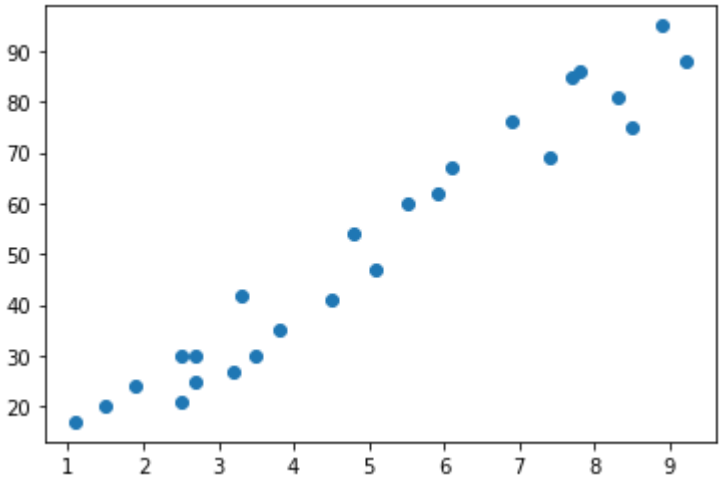
TRAINING

In [21]:
```
regressor.fit(x_train,y_train)
```

Out[21]: LinearRegression()

Plotting regression line

In [23]:
```
line=regressor.coef_*X+regressor.intercept_
plt.scatter(X,Y)
plt.show()
```



In [33]:
```
print(x_test)
y_pred=regressor.predict(x_test)
```

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

In [35]:
```
df=pd.DataFrame({'Actual':y_test,'Predicted':y_pred})
df.head()
```

Out[35]:

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

In [39]:
```
from sklearn.metrics import mean_absolute_error
MAE=mean_absolute_error(y_pred,y_test)
print('meanabsoluteError={}'.format(MAE))
```

```
meanabsoluteError=4.183859899002975
```

EVALUATING

In [40]:
```
hours=9.25
hours_arr=np.array(hours).reshape(-1,1)
predict_score=regressor.predict(hours_arr)
print("NO of hours={}".format(hours))
print('predicted score={}'.format(predict_score[0]))
```

```
NO of hours=9.25
predicted score=93.69173248737538
```