Prediction of percentage of marks that a student is expected to score based upon the number of hours they studied using LINEAR REGRESSION.

In [3]:
```
# STEP : 1 (importing libraries)
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [4]:
```
#STEP 2 : (importing dataset)
data = "http://bit.ly/w-data"
data_set = pd.read_csv(data)
print("First 10 element of given dataset")
data_set.head(10)
```

```
First 10 element of given dataset
```

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

Preparing Data

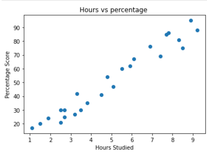
In [6]:
```
#STep 3: seperating dependent and independent variables:
x =data_set.iloc[:,0:1].values
x
```

Out[6]:
```
array([[2.5],
       [5.1],
       [3.2],
       [8.5],
       [3.5],
       [1.5],
       [9.2],
       [5.5],
       [8.3],
       [2.7],
       [7.7],
       [5.9],
       [4.5],
       [3.3],
       [1.1],
       [8.9],
       [2.5],
       [1.9],
       [6.1],
       [7.4],
       [2.7],
       [4.8],
       [3.8],
       [6.9],
       [7.8]])
```

In [7]:
```
y = data_set.iloc[0:,1:2].values
y
```

Out[7]:
```
array([[21],
       [47],
       [27],
       [75],
       [30],
       [20],
       [88],
       [60],
       [81],
       [25],
       [85],
       [62],
       [41],
       [42],
       [17],
       [95],
       [30],
       [24],
       [67],
       [69],
       [30],
       [54],
       [35],
       [76],
       [86]], dtype=int64)
```

In [8]:
```
#step 5: Scatterplot between independent and dependent variables
plt.scatter(x,y)
plt.title("Hours vs percentage ")
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



In [9]:
```
#step 6: seperating train and test sets
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)
x_train
```

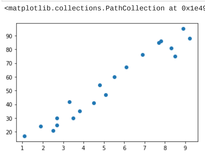
Out[9]:
```
array([[3.8],
       [1.9],
       [7.8],
       [6.9],
       [1.1],
       [5.1],
       [7.7],
       [3.3],
       [8.3],
       [9.2],
       [6.1],
       [3.5],
       [2.7],
       [5.5],
       [2.7],
       [8.5],
       [2.5],
       [4.8],
       [8.9],
       [4.5]])
```

In [10]:
```
y_train
```

Out[10]:
```
array([[35],
       [24],
       [86],
       [76],
       [17],
       [47],
       [85],
       [42],
       [81],
       [88],
       [67],
       [30],
       [25],
       [60],
       [30],
       [75],
       [21],
       [54],
       [95],
       [41]], dtype=int64)
```

In [11]:
```
plt.scatter(x_train,y_train)
```

Out[11]:
```
<matplotlib.collections.PathCollection at 0x1e49aa3c940>
```



Training the Algorithm

In [12]:
```
#Step 7: Linear Regression
from sklearn.linear_model import LinearRegression
linear_reg=LinearRegression()
linear_reg.fit(x_train,y_train)
print("Training complete ")
```

```
Training complete
```

Making Predictions

In [13]:
```
#Predicting the scores of the students
y_predict=linear_reg.predict(x_test)
y_predict
```

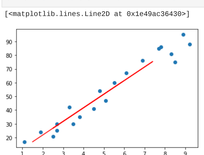
Out[13]:
```
array([[16.88414476],
       [33.73226078],
       [75.357018  ],
       [26.79480124],
       [60.49103328]])
```

In [14]:
```
#printing actual actual scores
y_test
```

Out[14]:
```
array([[20],
       [27],
       [69],
       [30],
       [62]], dtype=int64)
```

In [15]:
```
#Step 8: Checking the accuracy of the model
plt.scatter(x_train,y_train)
plt.plot(x_test, y_predict,color ="red")
```

Out[15]:
```
[<matplotlib.lines.Line2D at 0x1e49ac36430>]
```



In [16]:
```
#Step : 9 Testing data - In hours
print(x_test)
y_pred = linear_reg.predict(x_test)
```

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

In [17]:
```
# Step :10 The value of Predicted Scores if Student studies for 8.25 hours

linear_reg.predict([[9.25]])
```

Out[17]:
```
array([[93.69173249]])
```

Evaluating the model

In [18]:
```
#Step :11 Evaluating the model
from sklearn.metrics import r2_score
r2_score(y_test,y_predict)
```

Out[18]:
```
0.9454906892105356
```

In [ ]:
```

```