THE SPARKS FOUNDATION:DATA SCIENCE AND BUSINESS ANALYTICS

TASK1:PREDICTION USING SUPERVISED ML

AIM: predict the percentage of a students based on the number of study hours

LANGUAGE USED: python3

IDE: jupiter notebook

TYPE: Linear Regression

AUTHOR: KOTAPATI RAVICHANDRA

STEPS TO BE FOLLOWED

STEP1:import the dataset

STEP2:Vizualize and analyze the dataset

STEP3:Prepare the data

STEP4:Design and train the machine learing model

STEP5:Vizualize the model

STEP6:Make predictions

STEP7:Evaluate the model

STEP1:IMPORT THE DATA SET

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:

```
#reading the data from the link

data="http://bit.ly/w-data"
student_data=pd.read_csv(data)

print('data imported successfully')
student_data
```

```
data imported successfully
```

Out[2]:

	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]:

```
student_data.shape
```

Out[3]:

```
(25, 2)
```

In [5]:

```
student_data.describe()
```

Out[5]:

	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 [6]:

```
student_data.isnull().sum()
```

Out[6]:

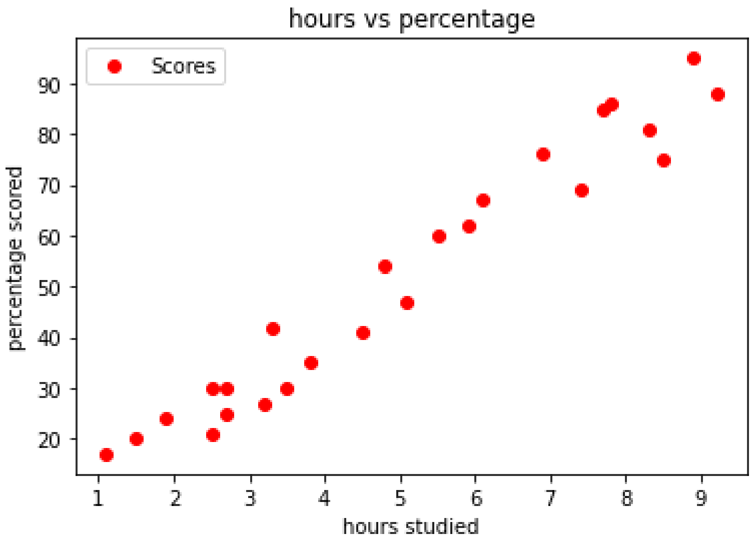
```
Hours     0
Scores    0
dtype: int64
```

STEP2:VISUALIZE AND ANALYZE THE DATASET

In [9]:

```
#ploting the distribution of scores and number of hours

student_data.plot(x='Hours',y='Scores',style='ro')
plt.title('hours vs percentage')
plt.xlabel('hours studied')
plt.ylabel('percentage scored')
plt.show()
```



STEP3:PREPARE THE DATA

In [13]:

```
x=student_data.iloc[:, :-1].values
y=student_data.iloc[:,  1].values
```

In [14]:

```
x
```

Out[14]:

```
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 [15]:

```
y
```

Out[15]:

```
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 [36]:

```
#now split the data into train and test data 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)
```

In [38]:

```
x_train
```

Out[38]:

```
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 [39]:

```

x_test
```

Out[39]:

```
array([[1.5],
       [3.2],
       [7.4],
       [2.5],
       [5.9]])
```

In [40]:

```
y_train
```

Out[40]:

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

In [41]:

```
y_test
```

Out[41]:

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

STEP4:DESIGN AND TRAIN THE MACHINE LEARNING MODEL

In [44]:

```
from sklearn.linear_model import LinearRegression

regressor=LinearRegression()
regressor.fit(x_train,y_train)

print("training complete")
```

```
training complete
```

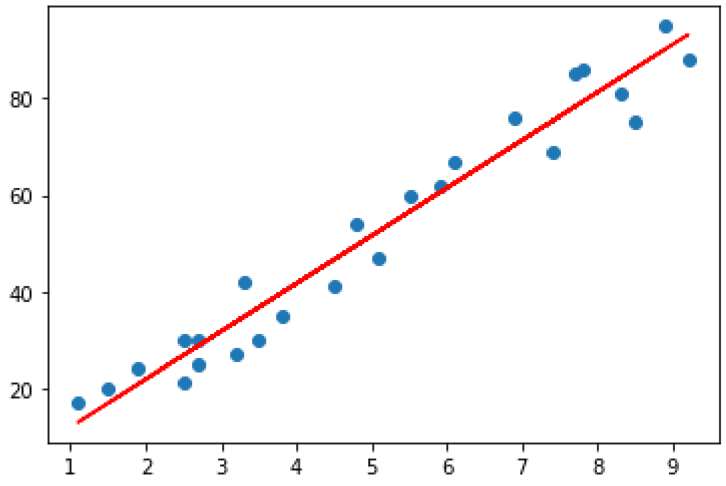
STEP5:VISUALIZE THE MODEL

In [47]:

```
#plotting reegression line

line=regressor.coef_*x+regressor.intercept_

#ploting for test data
plt.scatter(x,y)
plt.plot(x,line,color='red');
plt.show()
```



STEP6:MAKE PREDICTION

In [48]:

```
#making predictions
print(x_test)
y_pred=regressor.predict(x_test)
```

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

In [51]:

```
#predicted vs actual
df=pd.DataFrame({'Actual':y_test,'predicted':y_pred})
df
```

Out[51]:

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

In [53]:

```
#testing the custom input

hours=9.25
own_pred=regressor.predict([[hours]])
print(f"no of hours={hours}")
print(f"predicted score={own_pred[0]}")
```

```
no of hours=9.25
predicted score=93.69173248737539
```

STEP7:EVALUATE THE MODEL

In [54]:

```
#mean absolute error
from sklearn import metrics
print("mean absolute error:",metrics.mean_absolute_error(y_test,y_pred))
```

```
mean absolute error: 4.183859899002982
```

In [55]:

```
print("max error:",metrics.max_error(y_test,y_pred))
```

```
max error: 6.732260779489835
```

In [57]:

```
print("mean squared error:",metrics.mean_squared_error(y_test,y_pred))
```

```
mean squared error: 21.598769307217456
```

THANK YOU

In [ ]: