TASK 1: PREDICTION USING SUPERVISED ML

AUTHOR: VANIPENTA BALAJI GRIP JUNE 2023 THE SPARKS FOUNDATIONS

The Task is to predict the percentage of a student based on the number of study hours

I have used linear regression model for this task

In [1]:

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

In [2]:

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

data.head(10)
```

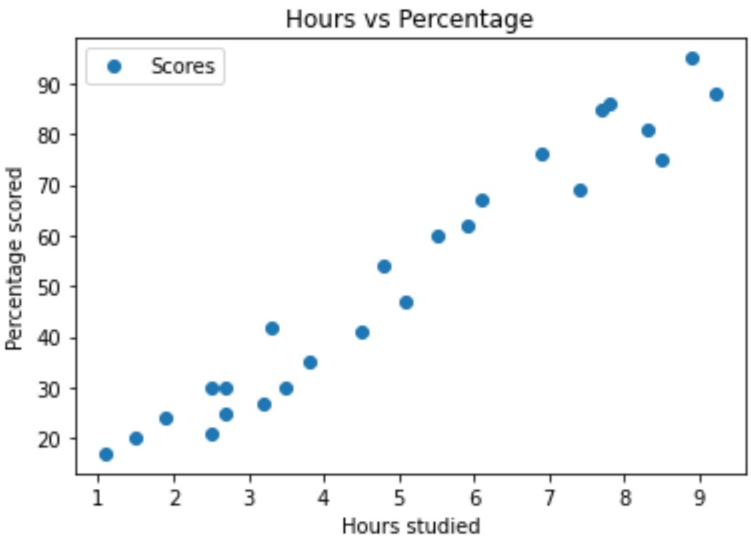
```
Data imported successfully
```

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

In [3]:

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



Preparing the data

Dividing the data into "attributes" (inputs) and "labels" (outputs)

In [4]:

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

In [5]:

```
#Split this data into training 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_st
```

Training the algorithm

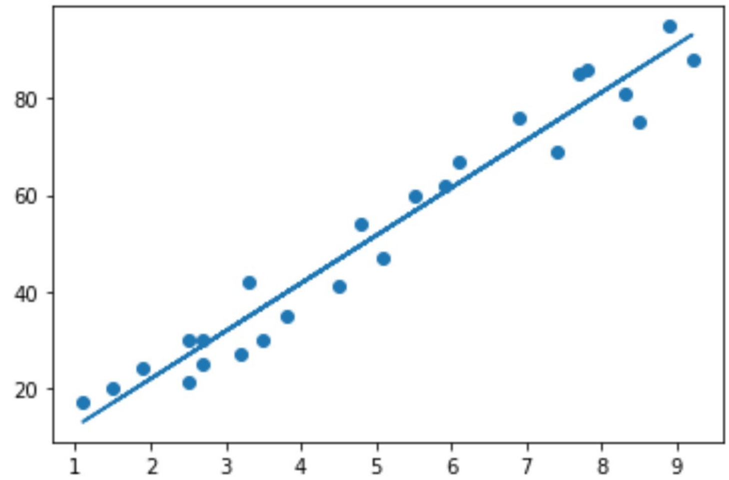
In [6]:

```
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(x_train,y_train)

print("Training Completed")
```

```
Training Completed
```

In [7]:

```
#Plotting the regression line

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



Making predictions

In [8]:

```
#Test data (Hours)
print(x_test)

#Predicting the scores
y_pred = regressor.predict(x_test)
```

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

In [9]:

```
y_pred
```

Out[9]:

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

In [10]:

```
y_test
```

Out[10]:

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

In [11]:

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

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

As per the task, testing with our own data

In [14]:

```
hours=9.25
own_pred = regressor.predict([[hours]])
print("No of Hours = {}".format([[hours]]))
print("Predicted Score = {}".format(own_pred[0]))
```

```
No of Hours = [[9.25]]
Predicted Score = 93.69173248737538
```

Evaluating the model

In [15]:

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

```
Mean Absolute Error: 4.183859899002975
```

In [ ]: