NAME : Deeksha Patil

Task 1 : Prediction Using Supervised ML

In this task we will be predicting the percentage of the students marks based on the number of hours.

First we need to import the following commands.

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Here I used the data from the URL

In [4]:
```
url = "http://bit.ly/w-data"
s_data = pd.read_csv(url)
print("Data imported successfully")

s_data.head(10)
```

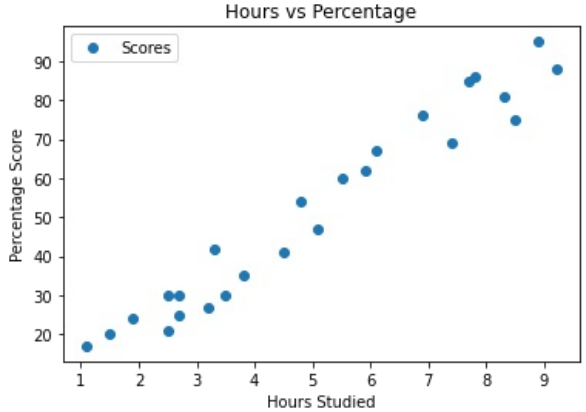
```
Data imported successfully
```

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

Then visualise for the input data. Plot the graph

In [5]:
```
s_data.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



It is observed that there is positive linear relation between the number of hours studied and the percentage of the scores

In [6]:
```
X = s_data.iloc[:, :-1].values
y = s_data.iloc[:, 1].values
```

Divide the data into "attributes" (inputs) and "labels" (outputs). Then split this data into training and test sets.

In [7]:
```
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)
```

Then we have to complete the algorithm training

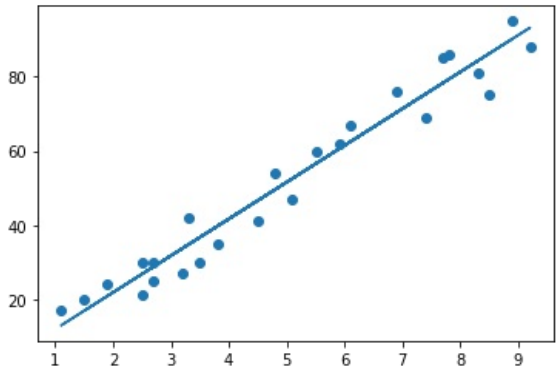
In [8]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

print("Training complete.")
```

```
Training complete.
```

Then plot the regression line for the test data.

In [9]:
```
line = regressor.coef_*X+regressor.intercept_
plt.scatter(X, y)
plt.plot(X, line);
plt.show()
```



Predictions:

In [10]:
```
#Predicting the scores
print(X_test)
#Testing data - In Hours
y_pred = regressor.predict(X_test) # Predicting the scores
```

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

In [11]:
```
#Comparision between the actual and predicted
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
```

```
df
```

Out[11]:

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

In [13]:

```
#You can also test with your own data
hours = 9.25
test=np.array([hours])
test=test.reshape(-1,1)
own_pred = regressor.predict(test)
print("No of Hours Studied= {}".format(hours))
print("Predicted Score = {}".format(own_pred[0]))
```

```
No of Hours Studied= 9.25
Predicted Score = 93.69173248737538
```

Evaluation of model using Mean Square Error

In [14]:

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

```
Mean Absolute Error: 4.183859899002975
```

In [ ]:

In [ ]: