Name: Vaishali - Prediction using Supervised ML(Task-1)

predict the percentage of marks of an student based on the number of study hours

In [3]:

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

In [5]:

```
# import cvs file from link
s_data = pd.read_csv("http://bit.ly/w-data")
s_data.head(24)
```

Out[5]:

	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

In [7]:

```
# check shape of Data
s_data.shape
```

Out[7]:

```
(25, 2)
```

In [9]:

```
s_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Hours   25 non-null     float64
 1   Scores  25 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

In [11]:

```
# check if any missing values are there
print("\nmissing values : ", s_data.isnull().values.sum())
```

```
missing values :  0
```

In [13]:

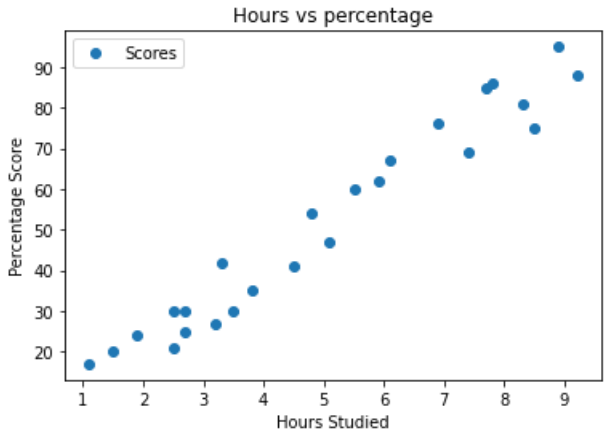
```
s_data.describe()
```

Out[13]:

	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 [15]:

```
# plotting the distribution of scores
s_data.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



In [17]:

```
# now divide the data into "attributes"(inputs) and "labels"(outputs)
x = s_data.iloc[:, :-1].values
y = s_data.iloc[:, 1].values
```

In [19]:

```
# then split the 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_state=0)
```

Training the Algorithm

In above coding we split our data now finally we train our algorithm.

In [21]:

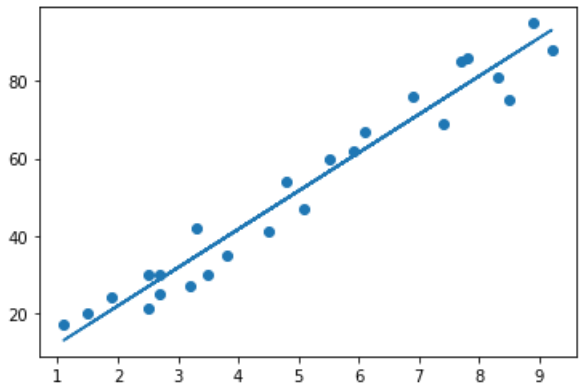
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train, y_train)
print("Training complete.")
```

```
Training complete.
```

In [22]:

```
# now we plot the regression line
line = regressor.coef_*x+regressor.intercept_

# then plotting for the test data
plt.scatter(x, y)
plt.plot(x, line);
plt.show()
```



Making Prediction

In above coding we train our algorithm now we make some predictions

In [24]:

```
# Time to make prediction first testing data in hours and second predicting the scores
print(x_test)
y_pred = regressor.predict(x_test)
```

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

In [26]:

```
# Comparing Actual vs predicting
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

Out[26]:

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

Task to predict the score for a student with study hours of 9.25

In [28]:

```
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 = 93.69173248737539
```

Evaluating the model

In [29]:

```
# Evaluating the model( evaluating the performance our algorithm)
from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
```

```
Mean Absolute Error: 4.183859899002982
```

THANK YOU