Name:Pranil Rego

Task 1 - Prediction using Supervised ML

Predict the percentage of an student based on the number of study hours.

Importing the libraries

In [1]:
```
#importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Data Reading

In [2]:
```
data = pd.read_csv('https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_score
print("Data imported successfully")

data.head(10)
```

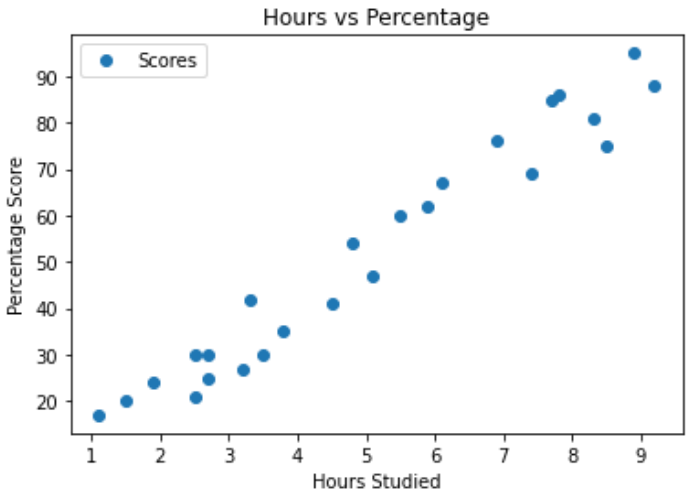
```
Data imported successfully
```

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

Plotting the Distrubution Of Scores

In [3]:
```
data.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



Dividing the Data

In [4]:
```
train = data.iloc[:, :-1].values
test = data.iloc[:, 1].values
```

Splitting the entire DataSet into Train and Test Data Set

In [5]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(train,test, test_size = 0.2, random_state = 0
```

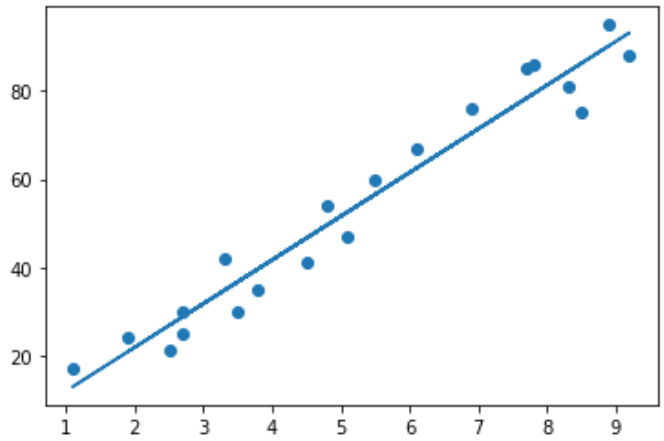
Training the Algorithm

In [6]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

print("Training complete.")
```

```
Training complete.
```

Plotting the regression line

In [13]:
```
# Plotting the regression line
line = regressor.coef_*X_train+regressor.intercept_

# Plotting for the test data
plt.scatter(X_train, y_train)
plt.plot(X_train, line);
plt.show()
```



Making Predictions

In [14]:
```
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]]
```

Comparing Actual vs Predicted

In [15]:
```
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

Out[15]:

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

Testing with Own Data

In [18]:
```
hours = 9.25
pred_score = Regressor1.predict([[9.25]])
print(" The Predicted Score of the Student is :",round(pred_score[0]))
```

```
 The Predicted Score of the Student is : 93
```

Evaluating the model

In [19]:
```
from sklearn import metrics
print('Mean Absolute Error:',
      metrics.mean_absolute_error(y_test, y_pred))
```

```
Mean Absolute Error: 4.183859899002975
```

In [ ]:
```

```