Author : Aryan Singh

Task 1 : Prediction using Supervised ML

GRIP @THE Sparks Foundation

Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

Importing Necessary Libraries

In [1]:
```
#importing all the essential libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

Step 1 : Loading the Dataset

In [2]:
```
#reading data from the url
url = "http://bit.ly/w-data"
data = pd.read_csv(url)
```

In [3]:
```
data.head(10)
```

Out[3]:

	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

Step 2 : Performing Exploratory Data Analysis

In [4]:
```
#preprocessing the data to apply the relevant model
x = data.iloc[:,:-1].values
```

```
y = data.iloc[:,1].values
print(x,y)
```

```
[[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]] [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]
```

Step 3 : Split Data into training and test dataset

In [5]:
```
#split the given data into training and test set
X_train, X_test, y_train, y_test = train_test_split(x, y,
                                   test_size=0.2, random_state=0)
print(X_train)
```

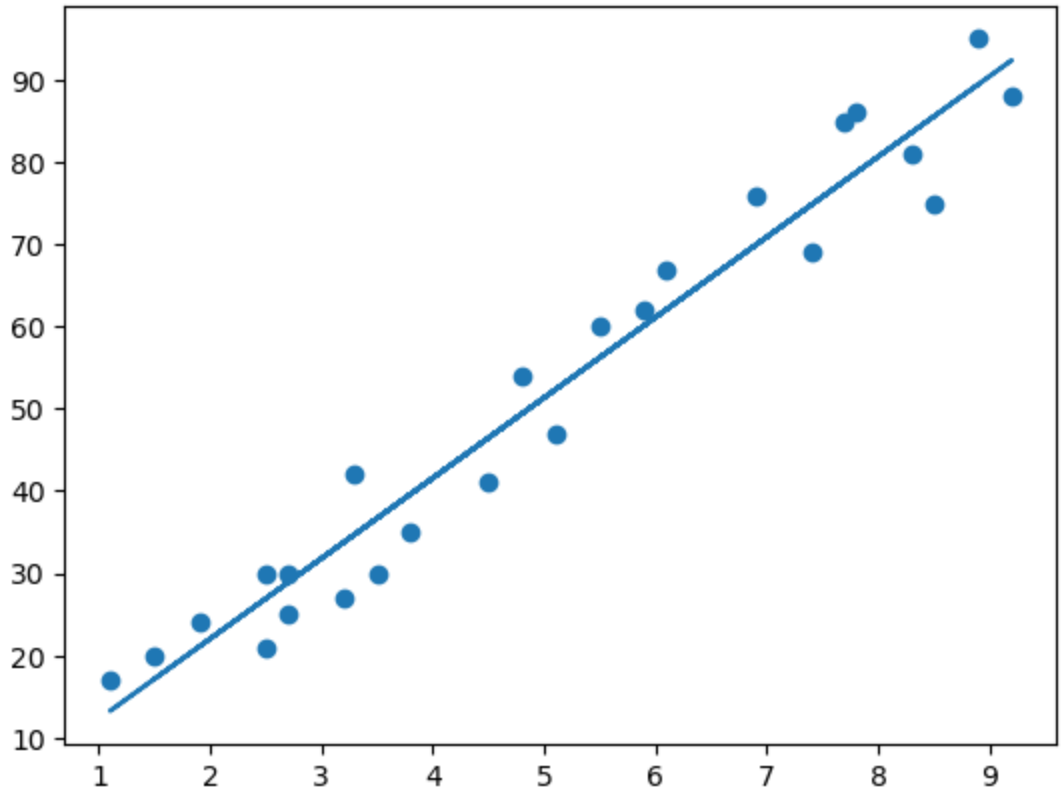
```
[[3.8]
 [1.9]
 [7.8]
 [6.9]
 [1.1]
 [5.1]
 [7.7]
 [3.3]
 [8.3]
 [9.2]
 [6.1]
 [3.5]
 [2.7]
 [5.5]
 [2.7]
 [8.5]
 [2.5]
 [4.8]
 [8.9]
 [4.5]]
```

Step 4 : Apply Linear Regression Model

In [6]:
```
model = LinearRegression().fit(x,y)
```

In [7]:
```
#Plot the regression line
```

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



In [8]:
```
#predicting scores using our test data
y_prediction = model.predict(X_test)
```

In [9]:
```
print(y_prediction)
```

```
[17.14737849 33.76624426 74.8246185  26.92318188 60.16091341]
```

Step 5 : Comparing the actual and predicted values

In [10]:
```
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_prediction})
df
```

Out[10]:

	Actual	Predicted
0	20	17.147378
1	27	33.766244
2	69	74.824618
3	30	26.923182
4	62	60.160913

In [11]:
```
#predicting the value for custom data
value = np.array([9.25])
predict = model.predict(value.reshape((-1,1)))
```

In [12]:
```
print(predict)
```

```
[92.90985477]
```

Step 6 : Evaluating the model

In [13]:
```
# Evaluating the model using mean square error
mean_squared_error(y_test,y_prediction)
```

Out[13]:
```
20.138948129940175
```

Conclusion

I was able to successfully carry out prediction using Supervised Machine Learning Algorithm and was also able to evaluate the model's accuracy score