Author : Muhammad Bilal

Prediction using Supervised ML

Problem Statement : Predict the percentage of an student based on the no. of study hours.What will be predicted score if a student studies for 9.25 hrs/ day?

Importing all libraries

In [27]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.metrics import mean_absolute_error
```

importing data

In [29]:
```
link = "http://bit.ly/w-data"
data = pd.read_csv(link)
print("Data imported successfully")
print(data)
```

```
Data imported successfully
    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
24    7.8      86
```

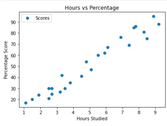
Checking for missing values

In [28]:
```
data.isna().sum
```

Out[28]:
```
<bound method DataFrame.sum of      Hours  Scores
0    False   False
1    False   False
2    False   False
3    False   False
4    False   False
5    False   False
6    False   False
7    False   False
8    False   False
9    False   False
10   False   False
11   False   False
12   False   False
13   False   False
14   False   False
15   False   False
16   False   False
17   False   False
18   False   False
19   False   False
20   False   False
21   False   False
22   False   False
23   False   False
24   False   False>
```

Plotting the Data

In [30]:
```
data.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



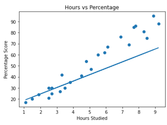
In [31]:
```
#splitting training and testing data
x=data.iloc[:,:-1].values
y=data.iloc[:,1].values
x_train, x_test, y_train, y_test= train_test_split(x, y, train_size=0.08, test_size=0.20, random_state=0)
```

In [32]:
```
from sklearn.linear_model import LinearRegression
linearRegressor= LinearRegression()
linearRegressor.fit(x_train,y_train)
y_predict= linearRegressor.predict(x_train)
```

In [33]:
```
regressor = LinearRegression()
regressor.fit(x_train, y_train)
print ("Training complete.")
```

```
Training complete.
```

In [34]:
```
#plotting the regression line
line = regressor.coef_*x+regressor.intercept_
#plotting for the test data
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.scatter(x,y)
plt.plot(x, line);
plt.show()
```



In [35]:
```
x = data.iloc[:,:-1].values
y = data.iloc[:,1].values
print(x)
print(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]
 [8.9]
 [4.5]]
[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]
```

Testing and Training

In [36]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.40, random_state = 0)
```

In [37]:
```
print(x_train)
print(x_test)
```

```
[[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]]
[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]
 [3.8]
 [1.9]
 [7.8]
 [6.9]
 [1.1]]
```

In [38]:
```
print(y_train)
print(y_test)
```

```
[47 85 42 81 88 67 30 25 60 30 75 21 54 95 41]
[20 27 69 30 62 35 24 86 76 17]
```

In [39]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train, y_train)
```

Out[39]:
```
LinearRegression()
```

In [40]:
```
y_pred = regressor.predict(x_test)
print('Predicted data\n' , y_pred)
```

```
Predicted data
 [15.9477618  32.773947423 74.344523   25.84551793 59.49788879 38.71260091
 19.90086425 78.303362546 69.39564493 11.98865934]
```

predicted values and actual values

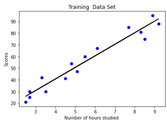
In [41]:
```
df = pd.DataFrame({'Predicted values':y_pred,'Actual values':y_test})
df
```

Out[41]:

	Predicted values	Actual values
0	15.947762	20
1	32.773947	27
2	74.344523	69
3	25.845518	30
4	59.497889	62
5	38.712601	35
6	19.906864	24
7	78.303625	86
8	69.395645	76
9	11.988659	17

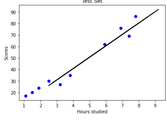
In [42]:
```
plt.scatter(x_train,y_train,color='BLUE')
plt.plot(x_train,regressor.predict(x_train),color='BLACK')
plt.xlabel("Number of hours studied ")
plt.ylabel("Scores")
plt.title("Training  Data Set")
```

Out[42]:
```
Text(0.5, 1.0, 'Training  Data Set')
```



In [43]:
```
plt.scatter(x_test,y_test,color='blue')
plt.plot(x_train,regressor.predict(x_train),color='black')
plt.xlabel("Hours studied")
plt.ylabel("Scores")
plt.title("Test Set")
```

Out[43]:
```
Text(0.5, 1.0, 'Test Set')
```



Prediction for 9.25 hours

In [44]:
```
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 = 92.65537184734602
```

From the above test we can predict that if a student studies for 9.25 hrs/day then he/she might score 92.66