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

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

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

Checking for missing values

data.isna().sum

Plotting the Data

data.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()

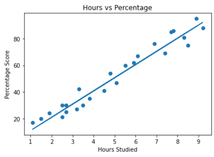
In [4]:
```
#splitting training and testing data
x=data.iloc[:,:-1].values
y=data.iloc[:,1].values
x_train, x_test, y_train, y_test= train_test_split(x, y, train_size=0.68, test_size=0.20, random_state=0)
```

In [5]:
```
from sklearn.linear_model import LinearRegression
linearRegressor= LinearRegression()
linearRegressor.fit(x_train,y_train)
y_predict= linearRegressor.predict(x_train)
```

In [6]:
```
regressor = LinearRegression()
regressor.fit(x_train, y_train)
print ("Training complete.")
```

```
Training complete.
```

In [16]:
```
#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 [8]:
```
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]
 [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]
```

Testing and Training

In [9]:
```
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 [10]:
```
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 [11]:
```
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 [12]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train, y_train)
```

Out[12]:
```
LinearRegression()
```

In [13]:
```
y_pred = regressor.predict(x_test)
print('Predicted data\n' , y_pred)
```

```
Predicted data
 [15.9477618  32.77394723 74.344523   25.84551793 59.49788879 38.71260091
 19.90686425 78.30362545 69.39564493 11.98865934]
```

predicted values and actual values

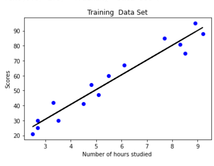
In [17]:
```
df = pd.DataFrame({'Predicted values':y_pred,'Actual values':y_test})
df
```

Out[17]:

	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

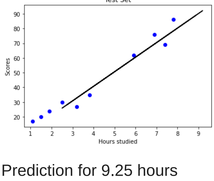
In [22]:
```
plt.scatter(x_train,y_train,color='BLUE')
plt.plot(x_train,regressor.predict(x_train),color='BLACK')
plt.xlabel("Number of hours studied ")
plt.ylabel("Scores")
plt.title("Training  Data Set")
```

Out[22]:
```
Text(0.5, 1.0, 'Training  Data Set')
```



In [23]:
```
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[23]:
```
Text(0.5, 1.0, 'Test Set')
```



Prediction for 9.25 hours

In [24]:
```
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

In [ ]:
```
```