Task 1 : Use Simple Linear Regression for fitting the best model for the given data set

AUTHOR NAME : THEJAS ELDHOS KURIAN

DATASET : http://bit.ly/w-data

Predict the score of student who works 9.25 hours/day

Step 1 : Importing neccessary libraries/packages

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
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,r2_score,mean_absolute_error
```

Step 2 : Importing the dataset

In [2]:
```
URL = "http://bit.ly/w-data"
data = pd.read_csv(URL)
data.head(10)
```

Out[2]:

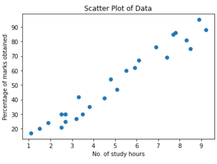
	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25

Step 3 : Choosing the appropriate variables

In [3]:
```
# x denotes independent variable
x = data["Hours"]
# y denotes egendent variable
y = data["Scores"]
```

Step 4 : Plotting the graph of given data

In [4]:
```
plt.scatter(x,y,marker = 'o')
plt.title("Scatter Plot of Data")
plt.xlabel("No. of study hours")
plt.ylabel("Percentage of marks obtained")
plt.show()
```



Step 5 : Splitting the data into testing and training data

In [5]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,train_size = 0.7 , random_state = 45)
```

Step 6 : Building the model with training data

In [6]:
```
model = LinearRegression().fit(np.array(x_train).reshape(-1,1),np.array(y_train).reshape(-1,1))
```

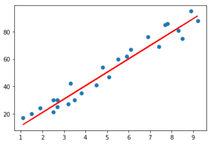
In [7]:
```
print("Model Coefficient")
print("Intercept : ",float(model.intercept_))
print("Coefficient : " ,float(model.coef_))
```

```
Model Coefficient
Intercept :  1.2162641848495284
Coefficient :  9.810924136707005
```

Step 7 : Plotting the Regression line in the scatter plot

In [8]:
```
line = float(model.intercept_) + x*float(model.coef_)
```

In [9]:
```
plt.scatter(x,y)
plt.plot(x,line,color = "red")
plt.show()
```



Step 8 : Making prediction on the test data

In [10]:
```
y_predicted = float(model.intercept_) + x_test*float(model.coef_)
y_predicted
```

Out[10]:
```
16    25.743575
20    27.705759
13    33.592314
10    76.760380
17    19.857020
24    77.741472
22    38.497776
19    73.817103
Name: Hours, dtype: float64
```

Another method for predicting values from Model created

In [11]:
```
y_predict_1 = model.predict(np.array(x_test).reshape(-1,1))
y_predict = pd.DataFrame(y_predict_1 , columns=["y_predicted"])
y_predict
```

Out[11]:

	y_predicted
0	25.743575
1	27.705759
2	33.592314
3	76.760380
4	19.857020
5	77.741472
6	38.497776
7	73.817103

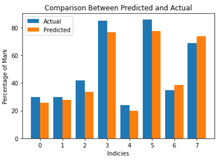
Step 9 : Comparing the predicted and test data set for evalutaing the model

In [12]:
```
dataframe = pd.DataFrame({"Actual":y_test,"Predicted":y_predicted})
dataframe.reset_index().drop(columns = ["index"])
```

Out[12]:

	Actual	Predicted
0	30	25.743575
1	27	27.705759
2	42	33.592314
3	85	76.760380
4	24	19.857020
5	86	77.741472
6	35	38.497776
7	69	73.817103

In [13]:
```
x_axis = np.arange(8)
plt.bar(x_axis-0.2,dataframe["Actual"],0.4,label = "Actual")
plt.bar(x_axis+0.2,dataframe["Predicted"],0.4,label = "Predicted")
plt.legend(loc = "upper left")
plt.title("Comparison Between Predicted and Actual")
plt.xlabel("Indicies")
plt.ylabel("Percentage of Mark")
plt.show()
```



Step 10 : Prediction using given data

In [14]:
```
x_new = np.array([9.25])
y_predict_new = float(model.intercept_) + x_new*float(model.coef_)
print("Number of Hours =",float(x_new))
print("Predicted percentage of score =",float(y_predict_new))
```

```
Number of Hours = 9.25
Predicted percentage of score = 91.96731244938931
```

Step 11 : Importance statistics relating to Model

In [15]:
```
m_a_s = mean_absolute_error(y_test,y_predicted)
print("Mean Absolute Error = ", m_a_s)
m_s_e = mean_squared_error(y_test,y_predicted)
print("Mean Squared Error = ", m_s_e)
print("Root Mean Squared Error = ",np.sqrt(m_s_e))
r2 = r2_score(y_test,y_predicted)
print("Coefficient of Determination = ",r2)
```

```
Mean Absolute Error =  5.489294806376118
Mean Squared Error =  35.34596219070804
Root Mean Squared Error =  5.945247052510213
Coefficient of Determination =  0.9391488478760097
```

Since coefficient of determination is 0.94 approximately.The model which we build best fits the data

Conclusion

I was successfully able to carry out the Prediction using Supervised Machine Learning Model.And evaluate model's performance using various parameters.

THANK YOU