The Sparks Foundation - # Aug21

Suhasi Parlikar

Task 1 : Prediction using supervised ML

Simple Linear Regression In this regression task we will predict the percentage of marks that a student is expected to score based upon the number of hours they studied. This is a simple linear regression task as it involves just two variables.

In [6]:
```
# Importing the libraries required for the task
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
```

In [7]:
```
# Reading the data
url='http://bit.ly/w-data'
data=pd.read_csv(url)
print("we have succesfully imported the data")
# printing the first 10 elements.
data.head(10)
```

```
we have succesfully imported the data
```

Out[7]:

	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

In [8]:
```
# shape of the dataset
data.shape
```

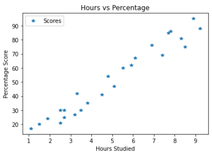
Out[8]:
```
(25, 2)
```

In [9]:
```
# Describing the data
data.describe()
```

Out[9]:

	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

In [10]:
```
# Plotting the distribution of scores
data.plot(x='Hours', y='Scores', style='*')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



Extracting Independent and Dependent Variable

In [11]:
```
# x is independent variable and y is dependent
# iloc[] indexing of dataset
X = data.iloc[: , :-1].values
y = data.iloc[: ,  1].values
print(X)
print()
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]
```

Spliting the data into traning and test set

In [12]:
```
# train_test_split randomly shuffles and divides all the rows

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                            test_size=0.25, random_state=42)
```

Training and implementing the model

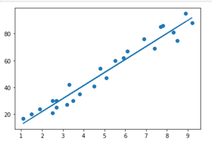
In [13]:
```
# Importing the linear regression model
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

print("Training complete.")
```

```
Training complete.
```

In [14]:
```
# Plotting the regression line
line = regressor.coef_*X+regressor.intercept_

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



In [15]:
```
print(X_test)
```

```
[[8.3]
 [2.5]
 [2.5]
 [6.9]
 [5.9]
 [2.7]
 [3.3]]
```

In [16]:
```
 # prediction of values
y_pred = regressor.predict(X_test)
print(y_pred)
```

```
[83.10733229 26.76559757 26.76559757 69.50760322 59.79351103 28.70841601
 34.5368713 ]
```

Comparing

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

Out[17]:

	Actual	Predicted
0	81	83.107332
1	30	26.765598
2	21	26.765598
3	76	69.507603
4	62	59.793511
5	25	28.708416
6	42	34.536871

Model evaluation

In [18]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error,r2_score
print("Mean Absolute Error:" , mean_absolute_error(y_test,y_pred))
print("mean squared Error" , mean_squared_error(y_test,y_pred))
print("R^2 score:", r2_score(y_test,y_pred))
```

```
Mean Absolute Error: 4.425394675156183
mean squared Error 23.516396034595225
R^2 score: 0.9553509219739938
```

In [21]:
```
hours = float(input("enter the hour"))
own_pred = regressor.predict([[hours]])
print("predicted percentage: %.2f"%(own_pred))
#print("No of Hours = {}".format(hours))
#print("Predicted Score = {}".format(own_pred[0]))
```

```
enter the hour9.5
predicted percentage: 94.76
```

In [20]:
```
from sklearn import metrics
print('Mean Absolute Error:',
      metrics.mean_absolute_error(y_test, y_pred))
```

```
Mean Absolute Error: 4.425394675156183
```

Conclusion:

1) A student will score 80.19 % if he/she study for 8 hours

2) The r^2 value for this Linear Regression model is 0.955