GRIP : The Sparks Foundation

Data Science and Business Analytics Intern

Author : Neha Jha

Task 1 : Prediction using Supervised ML

In this task we have to predict the percentage score of a student based on the number of hours studied. The task has two variables where the feature is the no. of hours studied and the target value is the percentage score. This can be solved using simple linear regression.

In [14]:

```
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.linear_model import LinearRegression
```

In [5]:

```
df = pd.read_csv("http://bit.ly/w-data")
df
```

Out[5]:

	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

In [6]:

```
df.head()
```

Out[6]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

EDA

In [38]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Hours   25 non-null     float64
 1   Scores  25 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

In [7]:

```
df.columns
```

Out[7]:

```
Index(['Hours', 'Scores'], dtype='object')
```

In [8]:

```
df.dtypes
```

Out[8]:

```
Hours     float64
Scores      int64
dtype: object
```

In [9]:

```
df.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]:

```
df.isnull().sum()
```

Out[10]:

```
Hours     0
Scores    0
dtype: int64
```

In [11]:

```
df.corr()
```

Out[11]:

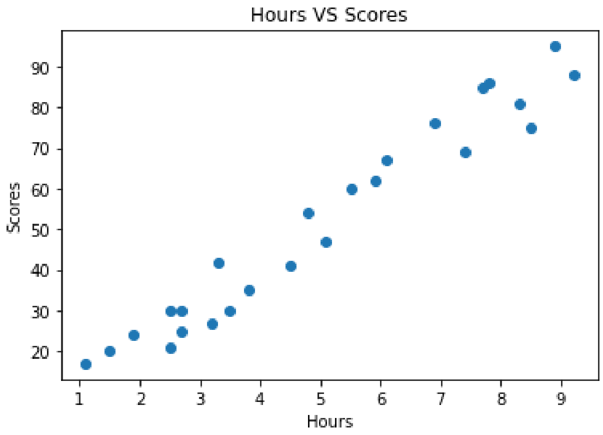
	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

In [12]:

```
plt.scatter(df['Hours'],df['Scores'])
plt.xlabel('Hours')
plt.ylabel('Scores')
plt.title('Hours VS Scores' )
```

Out[12]:

```
Text(0.5, 1.0, 'Hours VS Scores')
```



Linear Regression

In [18]:

```
x = df.iloc[:,:-1].values
y = df.iloc[:,1].values
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3,random_state = 0)
```

In [22]:

```
lnreg = LinearRegression()
lnreg
```

Out[22]:

```
LinearRegression()
```

In [23]:

```
lnreg.fit(x_train,y_train)
```

Out[23]:

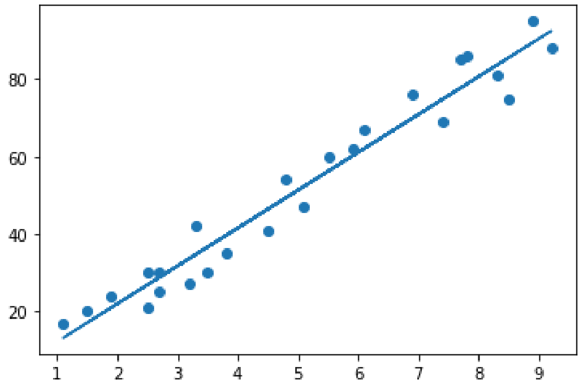
```
LinearRegression()
```

In [24]:

```
m=lnreg.coef_
c=lnreg.intercept_
l=m*x+c
plt.scatter(x,y)
plt.plot(x,l)
plt.show
```

Out[24]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [25]:

```
lnreg.intercept_
```

Out[25]:

```
2.3708153823418883
```

In [26]:

```
print(lnreg.coef_)
```

```
[9.78856669]
```

In [27]:

```
y_predict= lnreg.predict(x_test)
```

In [29]:

```
actual_predicted = pd.DataFrame({'Actual':y_test,'Predicted':y_predict})
actual_predicted
```

Out[29]:

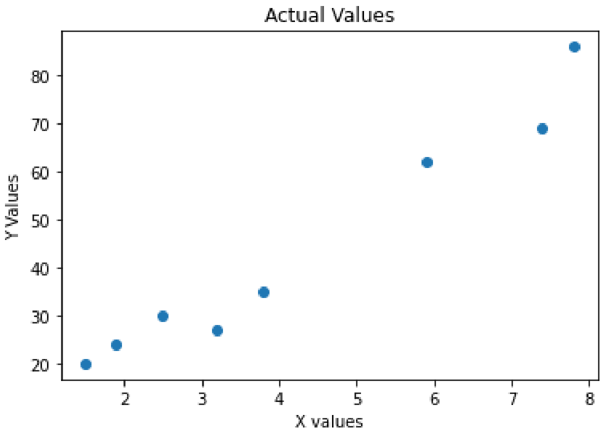
	Actual	Predicted
0	20	17.053665
1	27	33.694229
2	69	74.806209
3	30	26.842232
4	62	60.123359
5	35	39.567369
6	24	20.969092
7	86	78.721636

In [31]:

```
plt.scatter(x_test,y_test)
plt.xlabel('X values')
plt.ylabel('Y Values')
plt.title('Actual Values')
```

Out[31]:

```
Text(0.5, 1.0, 'Actual Values')
```

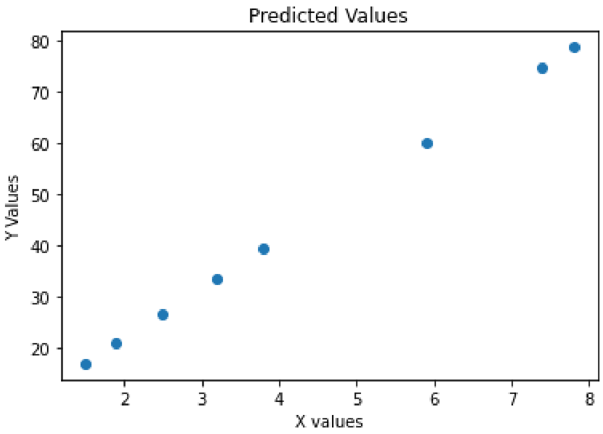


In [32]:

```
plt.scatter(x_test,y_predict)
plt.xlabel('X values')
plt.ylabel('Y Values')
plt.title('Predicted Values')
```

Out[32]:

```
Text(0.5, 1.0, 'Predicted Values')
```



what would be the predicted score if a student studies for 9.25 hrs/day ?

In [36]:

```
score= lnreg.predict([[9.25]])
print(score)
```

```
[92.91505723]
```

In [37]:

```
print("If a student studies for {} hours/day then he/she will score {} % in exam".format(9.25,score))
```

```
If a student studies for 9.25 hours/day then he/she will score [92.91505723] % in exam
```

Thank You