Prediction using Supervised Machine Learning using Simple Linear Regression

In this task we have to find the students scores based on their study hours. This is a simple Regression problem type because it has only two variables.

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
df=pd.read_csv('StudentHoursScores.csv')
df
```

Out[1]:

	Hours	Scores
0	7.7	79
1	5.9	60
2	4.5	45
3	3.3	33
4	1.1	12
5	8.9	87
6	2.5	21
7	1.9	19
8	2.7	29
9	8.3	81
10	5.5	58
11	9.2	88
12	1.5	14
13	3.5	34
14	8.5	85
15	3.2	32
16	6.5	66
17	2.5	21
18	9.6	96
19	4.3	42
20	4.1	40
21	3.0	30
22	2.6	25

In [2]:

```
df.head()

```

Out[2]:

	Hours	Scores
0	7.7	79
1	5.9	60
2	4.5	45
3	3.3	33
4	1.1	12

In [3]:

```
df.tail()
```

Out[3]:

	Hours	Scores
18	9.6	96
19	4.3	42
20	4.1	40
21	3.0	30
22	2.6	25

In [4]:

```
df.dtypes
```

Out[4]:

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

In [5]:

```
df.columns
```

Out[5]:

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

In [6]:

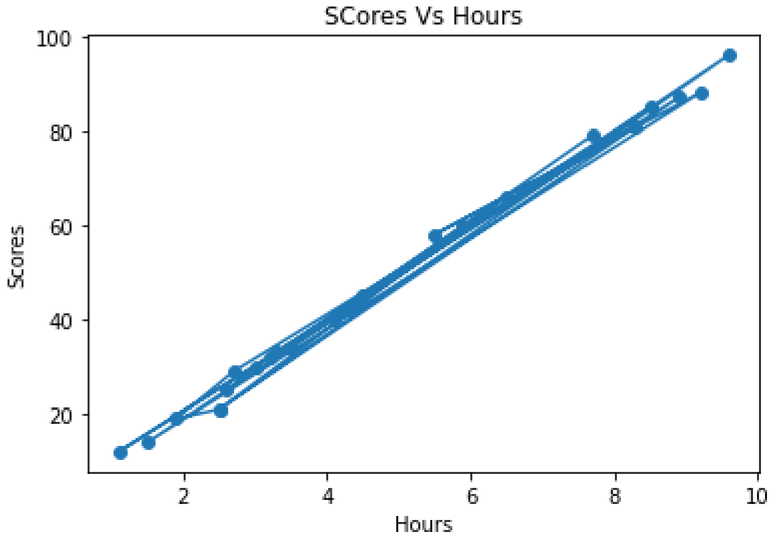
```python
df.shape
```

Out[6]:

```
(23, 2)
```

In [7]:

```python
plt.scatter(df.Hours,df.Scores)
plt.plot(df.Hours,df.Scores)
plt.xlabel('Hours')
plt.ylabel('Scores')
plt.title('SCores Vs Hours')
plt.show()
```



In [8]:

```python
df.describe()
```

Out[8]:

	Hours	Scores
count	23.000000	23.000000
mean	4.817391	47.695652
std	2.709688	27.103228
min	1.100000	12.000000
25%	2.650000	27.000000
50%	4.100000	40.000000
75%	7.100000	72.500000
max	9.600000	96.000000

In [9]:

```
df.min()
```

Out[9]:

```
Hours      1.1
Scores    12.0
dtype: float64
```

In [10]:

```
df.max()
```

Out[10]:

```
Hours      9.6
Scores    96.0
dtype: float64
```

In [11]:

```
df.corr()
```

Out[11]:

	Hours	Scores
Hours	1.000000	0.997656
Scores	0.997656	1.000000

Spliting of dataset into Testing and Training

In [12]:

```

x=df.iloc[:,:-1]
y=df.iloc[:,1]
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest=train_test_split(x,y,test_size=0.2,random_state=1)
```

Creating Simple Linear Model

In [13]:

```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(xtrain,ytrain)
```

Out[13]:

```
LinearRegression()
```

Prediction

In [14]:

```python
ypred=reg.predict(xtest)
reg.predict([[9.5]])
```

Out[14]:

```
array([94.37683212])
```

In [15]:

```python
ypred
```

Out[15]:

```
array([40.87711348, 25.025345  , 32.95122924, 34.9327003 , 42.85858454])
```

In [16]:

```python
ytest
```

Out[16]:

```
20    40
17    21
3     33
13    34
19    42
Name: Scores, dtype: int64
```

In [17]:

```python
xtest
```

Out[17]:

	Hours
20	4.1
17	2.5
3	3.3
13	3.5
19	4.3

In [18]:

```python
xpred=reg.predict([[5.2]])
```

In [19]:

```python
xpred
```

Out[19]:

```
array([51.77520431])
```

In [20]:

```
reg.coef_
```

Out[20]:

```
array([9.9073553])
```

In [21]:

```
reg.intercept_
```

Out[21]:

```
0.2569567372371182
```

In [22]:

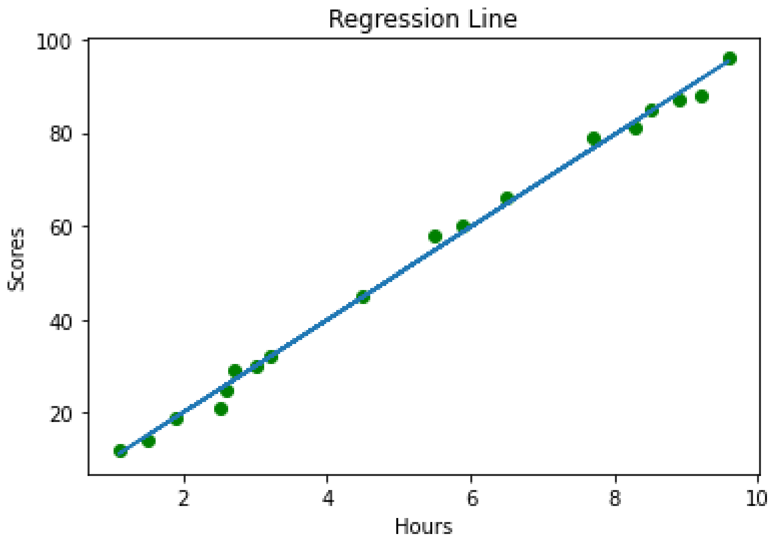
```
#y=mx+c

9.9073553*5.2 + 0.2569567372371182
```

Out[22]:

```
51.775204297237124
```

In [23]:

```
plt.scatter(xtrain,ytrain,color='g')
plt.plot(xtrain,reg.predict(xtrain))
plt.xlabel('Hours')
plt.ylabel('Scores')
plt.title('Regression Line')
plt.show()
```



In [24]:

```
print("Coefficient is:",reg.coef_)
print('Intercept is :',reg.intercept_)
```

```
Coefficient is: [9.9073553]
Intercept is : 0.2569567372371182
```

In [25]:

```
from sklearn.metrics import r2_score,mean_squared_error

accuracy=r2_score(xtest,ypred)
print("Accuracy of Model is :",accuracy)

```

```
Accuracy of Model is : -2564.9063623489724
```

In [26]:

```

ip = float(input("Enter no. of hours: "))
answer = reg.predict([[ip]])
print("Predicted score is:", answer)
```

```
Enter no. of hours: 5
Predicted score is: [49.79373325]
```

In [28]:

```
print("Mean Square Error is :", mean_squared_error(xtest,ypred))
```

```
Mean Square Error is : 1042.784345658622
```

In [ ]:

```

```