Prediction using Supervised ML

The Sparks Foundation

By-Sourav Srivastav

Import library

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
sns.set()
```

Import Data

In [2]:

```
df=pd.read_csv('https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_sc
df.head(15)
```

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
10	7.7	85
11	5.9	62
12	4.5	41
13	3.3	42
14	1.1	17

Summary of data(Mean,max,sd,etc)

In [3]:

```
df.describe()
```

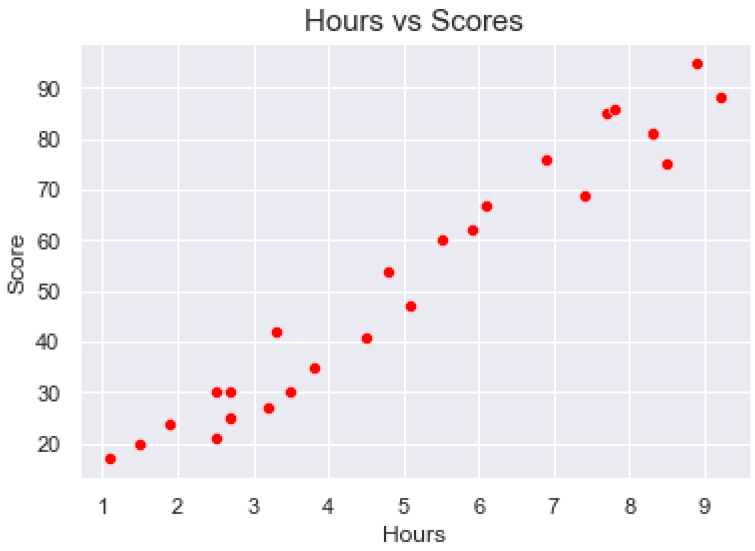
Out[3]:

	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

Required Plots

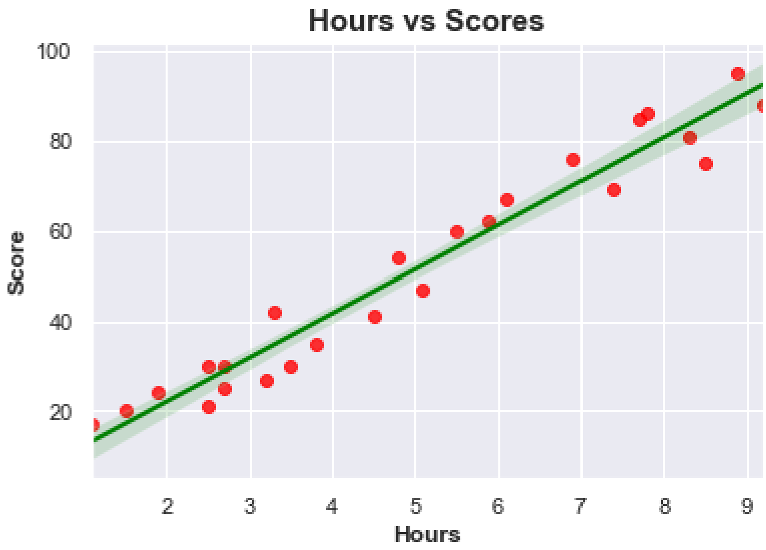
In [5]:

```
sns.scatterplot(x = df["Hours"],
 y = df["Scores"],
 color = "red")
plt.title("Hours vs Scores", fontsize =15)
plt.xlabel("Hours")
plt.ylabel("Score")
plt.show()
```



In [7]:

```
sns.regplot(x = df["Hours"],
 y = df["Scores"],
 data= df,
 scatter_kws = {'color' : "red"},
 line_kws = {'color': "green"})
plt.xlabel("Hours",fontweight = "bold")
plt.ylabel("Score",fontweight = "bold")
plt.title("Hours vs Scores", fontsize =15, fontweight ="bold")
plt.show()
```



Regression Analysis

In [8]:

```
X = df.iloc[:, :-1].values
y = df.iloc[:, 1].values
train_X, val_X, train_y, val_y = train_test_split(X, y, random_state = 0)
```

In [9]:

```
regression = LinearRegression()
regression.fit(train_X, train_y)
```

Out[9]:

```
LinearRegression()
```

In [10]:

```
print(val_X)
pred_y = regression.predict(val_X)
```

```
[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]
 [3.8]
 [1.9]]
```

In [11]:

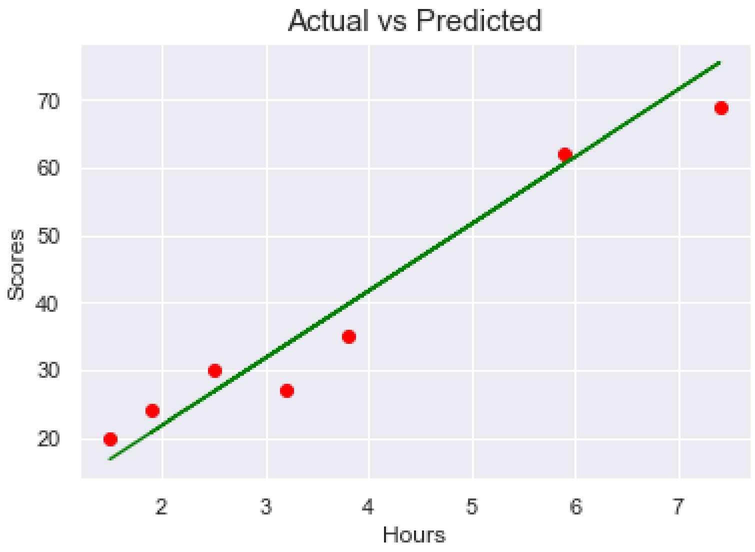
```
df1= pd.DataFrame({'Actual_Values': val_y, 'Predicted_Values': pred_y})
print(df1)
```

```
   Actual_Values  Predicted_Values
0             20         16.844722
1             27         33.745575
2             69         75.500624
3             30         26.786400
4             62         60.588106
5             35         39.710582
6             24         20.821393
```

Actual and Predicted Values on Plot

In [12]:

```
plt.scatter(x=val_X, y=val_y, color='red')
plt.plot(val_X, pred_y, color='green')
plt.title("Actual vs Predicted", fontsize=15)
plt.ylabel("Scores")
plt.xlabel("Hours")
plt.show()
```



In [15]:

```
print('Mean absolute error: ',mean_absolute_error(val_y,pred_y))
```

```
Mean absolute error:  4.130879918502486
```

In [16]:

```
hours = [9.25]
score = regression.predict([hours])
print("Score = {}".format(score))
```

```
Score = [93.89272889]
```

Conclusion

The predicted score if student studies for 9.5hrs/day is 93.89272889