Graduate Rotational Internship Program: The Sparks Foundation

Sunny Sanjivkumar Khade

Task 1 : Predict the percentage of an student based on the no. of study hours.

Simple Linear Regression (Using Python Scikit-Learn)

Importing Relevant Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()

from sklearn.linear_model import LinearRegression
```

Importing CSV file containing the data

In [5]:
```
data = pd.read_csv(r"E:\Sparks Internship Tasks\Study hours and Scores (Sparks Task 1) csv.csv")
data.head()
```

Out[5]:

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

In [6]:
```
data.describe()
```

Out[6]:

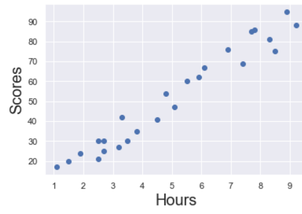
	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

Declaring Dependant and Independent variables

In [8]:
```
x = data['Hours']
y = data['Scores']
```

Plotting the distribution of scores

In [9]:
```
plt.scatter(x,y)
plt.xlabel('Hours', fontsize=20)
plt.ylabel('Scores', fontsize=20)
plt.show()
```



In [12]:
```
x_matrix = x.values.reshape(-1,1)
```

Splitting the data into Training and Testing sets

In [16]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x_matrix, y, test_size=0.2, random_state=0)
```

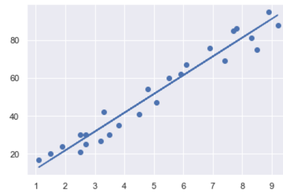
Training the algorithm (Linear Regression)

In [17]:
```
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(x_train, y_train)
```

Out[17]:
```
LinearRegression()
```

Plotting the regression line

In [18]:
```
line = regressor.coef_*x+regressor.intercept_
plt.scatter(x,y)
plt.plot(x,line)
plt.show()
```



Making predictions

In [19]:
```
print(x_test)
y_pred=regressor.predict(x_test)
```

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

In [20]:
```
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

Out[20]:

	Actual	Predicted
5	20	16.884145
2	27	33.732261
19	69	75.357018
16	30	26.794801
11	62	60.491033

Predicting score of a student who studies 9.25 hours per day

In [29]:
```
my_pred=regressor.predict(np.array([9.25]).reshape(1,1))
```

In [31]:
```
print('Predicted score for 9.25 hours per day= {}'.format(my_pred[0]))
```

```
Predicted score for 9.25 hours per day= 93.69173248737539
```

Evaluating the model (Mean Squared error)

In [33]:
```
from sklearn import metrics
mean_absolute_error=metrics.mean_absolute_error(y_test,y_pred)
print('Mean absolute error =',mean_absolute_error )
```

```
Mean absolute error = 4.183859899002982
```

In [ ]:
```

```