" THE SPARKS FOUNDATION "

DATA SCIENCE AND BUSINESS ANALYTICS

NAME- ASHRAF SHAIKH

TASK 1 : PREDICTION USING SUPERVISED MACHINE LEARNING

PROBLEM STATEMENT : The given dataset contains the score of students with respect to their study time. It is required to perform EDA and simple linear regression on the dataset to find out the score of a student who studies 9.5 hours/day

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv('http://bit.ly/w-data')
data.head(25)
```

```
    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
```

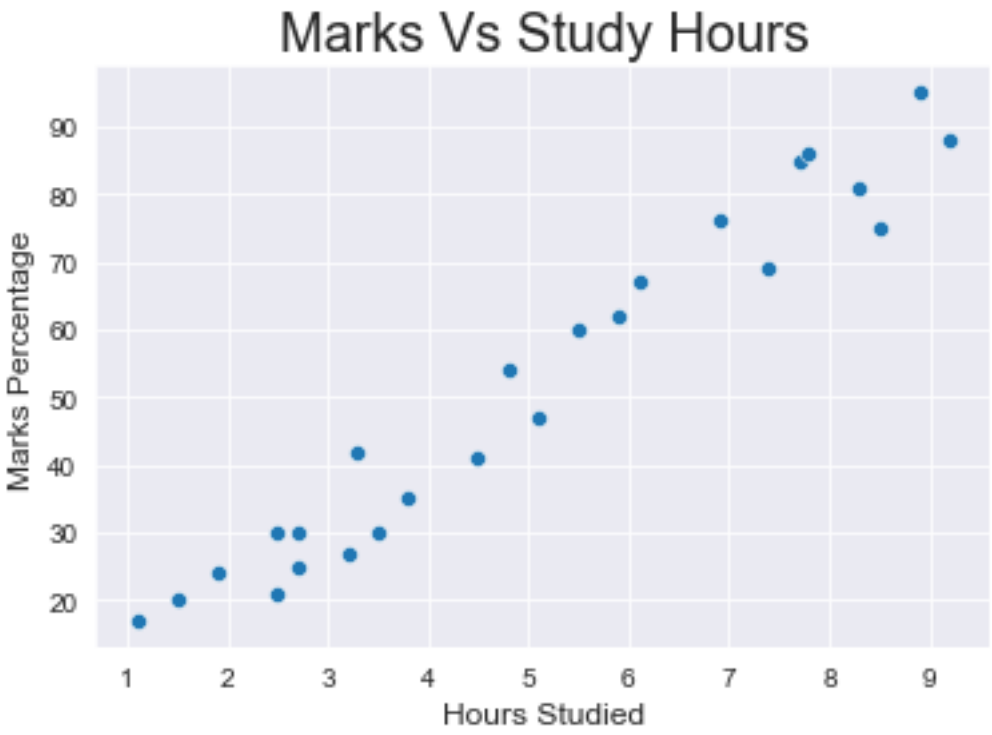
```
print("Successfully imported data " )
```

```
Successfully imported data
```

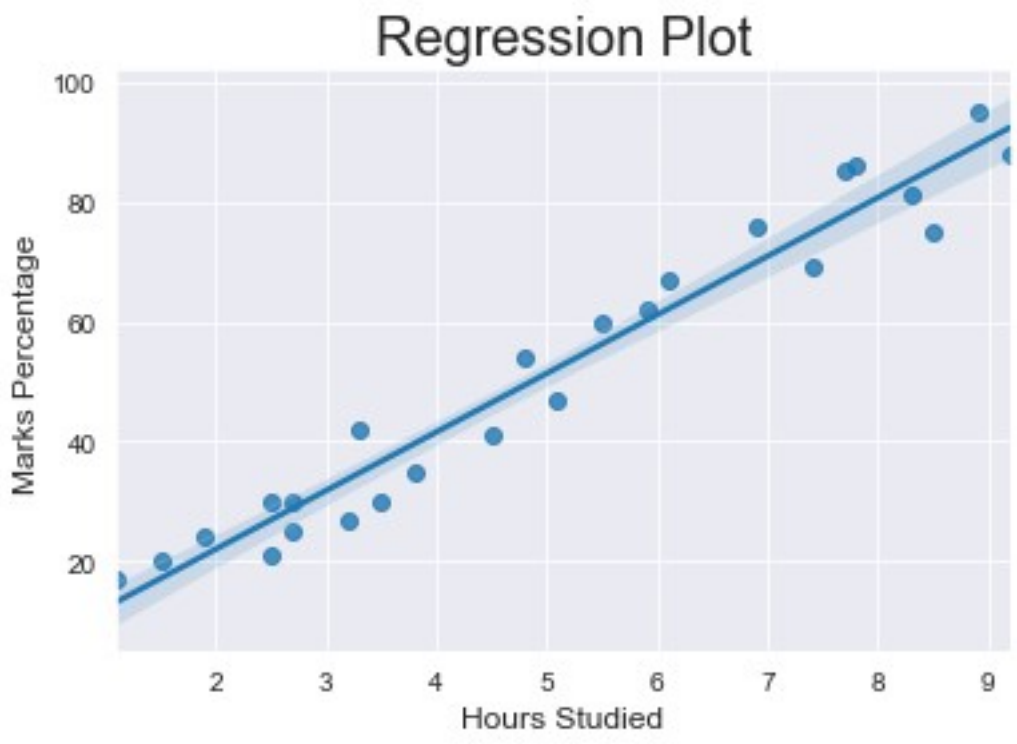
```
data.isnull == True
```

```
False
```

```
sns.set_style('darkgrid')
sns.scatterplot(y= data['Scores'], x= data['Hours'])
plt.title('Marks Vs Study Hours',size=20)
plt.ylabel('Marks Percentage', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
```




```
sns.regplot(x= data['Hours'], y= data['Scores'])
plt.title('Regression Plot',size=20)
plt.ylabel('Marks Percentage', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
print(data.corr())
```




```
            Hours    Scores
Hours    1.000000  0.976191
Scores   0.976191  1.000000
```

```
# Defining X and y from the Data
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values

# Spliting the Data in two
train_X, val_X, train_y, val_y = train_test_split(X, y, random_state =
0)

regression = LinearRegression()
regression.fit(train_X, train_y)
print("---------Model Trained---------")
```

```
---------Model Trained---------
```

```
pred_y = regression.predict(val_X)
prediction = pd.DataFrame({'Hours': [i[0] for i in val_X], 'Predicted
Marks': [k for k in pred_y]})
prediction
```

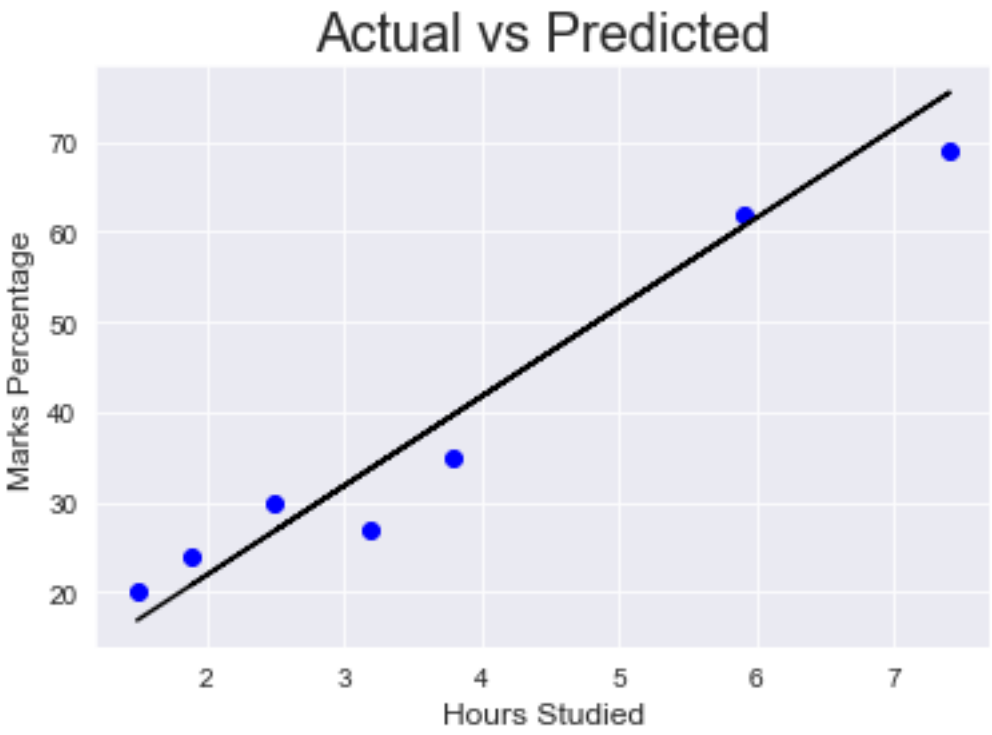
```
   Hours  Predicted Marks
0    1.5        16.844722
1    3.2        33.745575
2    7.4        75.500624
3    2.5        26.786400
```

```
4    5.9        60.588106
5    3.8        39.710582
6    1.9        20.821393
```

```
compare_scores = pd.DataFrame({'Actual Marks': val_y, 'Predicted
Marks': pred_y})
compare_scores
```

```
   Actual Marks  Predicted Marks
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
```

```
plt.scatter(x=val_X, y=val_y, color='blue')
plt.plot(val_X, pred_y, color='Black')
plt.title('Actual vs Predicted', size=20)
plt.ylabel('Marks Percentage', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
```




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

```
Mean absolute error:  4.130879918502482
```

```
hours = [9.25]
answer = regression.predict([hours])
print("Score = {}".format(round(answer[0],3)))
```

```
Score = 93.893
```

```
compare_scores .plot(kind='bar')
```

```
<AxesSubplot:>
```