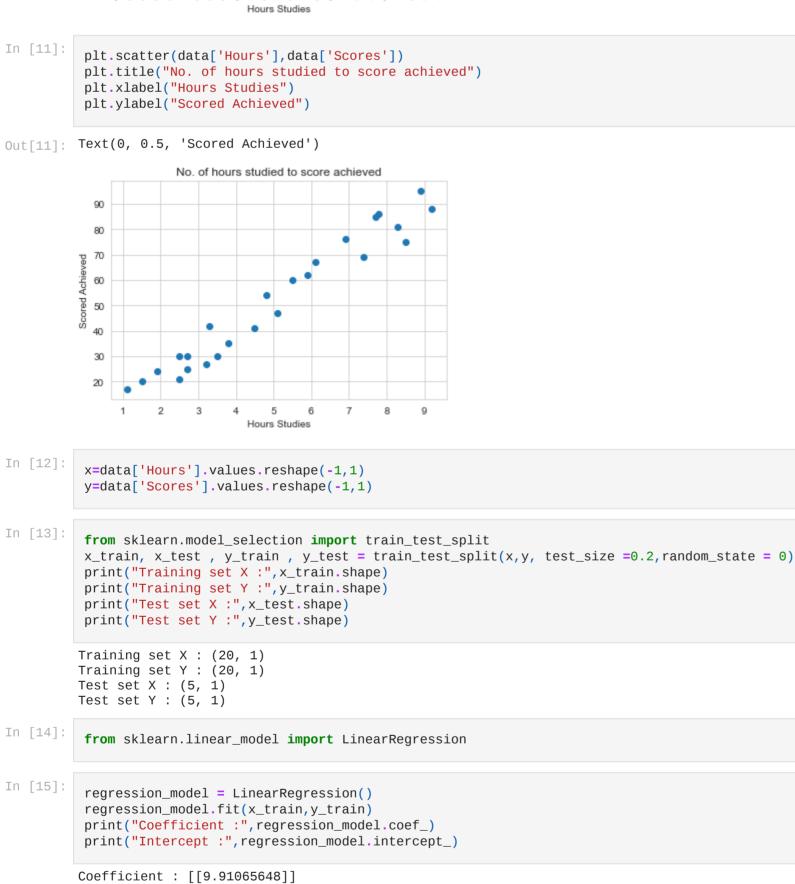
The Sparks Foundation Prediction of Scores Using Linear Regression Task 1 import pandas as pd import numpy as np import seaborn as sns import matplotlib.pyplot as plt %matplotlib inline In [2]: %pwd 'C:\\Users\\hp' In [3]: data = pd.read_csv("C:\\Users\\hp\\Desktop\\data.csv") data.head() Out[3]: **Hours Scores** 2.5 21 47 5.1 3.2 27 8.5 75 3.5 30 In [4]: data.shape Out[4]: (25, 2) data.dtypes float64 Hours int64 Scores dtype: object In [6]: data.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 data.describe() Out[7]: count 25.000000 25.000000 5.012000 51.480000 mean 2.525094 25.286887 1.100000 17.000000 min **25**% 2.700000 30.000000 **50**% 4.800000 47.000000 7.400000 75.000000 **75**% 9.200000 95.000000 max In [8]: fig, axes = plt.subplots(1,2,figsize=(12,4)) sns.set_style('whitegrid') sns.boxplot(ax=axes[0], x=data['Hours'], color='blue') sns.boxplot(ax=axes[1], x=data['Scores'], color='blue') plt.show() In [9]: corr=data.corr() print(corr) sns.heatmap(corr, annot=True, cmap="Blues") Scores Hours Hours 1.000000 0.976191 Scores 0.976191 1.000000 Out[9]: <AxesSubplot:> 1.000 - 0.995 0.98 - 0.990 - 0.985 0.98 - 0.980 Hours Scores In [10]: data.plot(kind = 'bar', x = 'Hours', y = 'Scores', style="o", color='orange', rot=60) plt.title("Hours Vs Scores") plt.xlabel("Hours Studies") plt.ylabel("Score Acheived") Out[10]: Text(0, 0.5, 'Score Acheived') Hours Vs Scores Scores 80 20 Hours Studies In [11]: plt.scatter(data['Hours'], data['Scores']) plt.title("No. of hours studied to score achieved") plt.xlabel("Hours Studies") plt.ylabel("Scored Achieved") Out[11]: Text(0, 0.5, 'Scored Achieved') No. of hours studied to score achieved 90



Intercept : [2.01816004]

plt.plot(x,line)

80

60

y_pred

In [18]:

In [19]:

In [20]:

In []:

In []:

Out[17]: array([[16.88414476],

Percentage

Out[20]: array([[93.7]])

plt.scatter(x,y,c="Black")
plt.title("Regression Plot")
plt.xlabel("Hours Studied")
plt.ylabel("Scores Achieved")

Out[16]: [<matplotlib.lines.Line2D at 0x1de6b5d1bb0>]

Regression Plot

y_pred = regression_model.predict(x_test)

print("R2 Score :",r2_score(y_test, y_pred))

Percentage = regression_model.predict(Hour)
Percentage = np.round(Percentage, decimals=1)

Mean Absolute Error : 4.183859899002975

Hour = np.array([9.25]).reshape(-1,1)

R2 Score : 0.9454906892105356

from sklearn.metrics import mean_absolute_error ,r2_score

print("Mean Absolute Error :", mean_absolute_error(y_test, y_pred))

[33.73226078], [75.357018], [26.79480124], [60.49103328]])

line = regression_model.coef_*x + regression_model.intercept_

In [16]:

Pranjal Jain