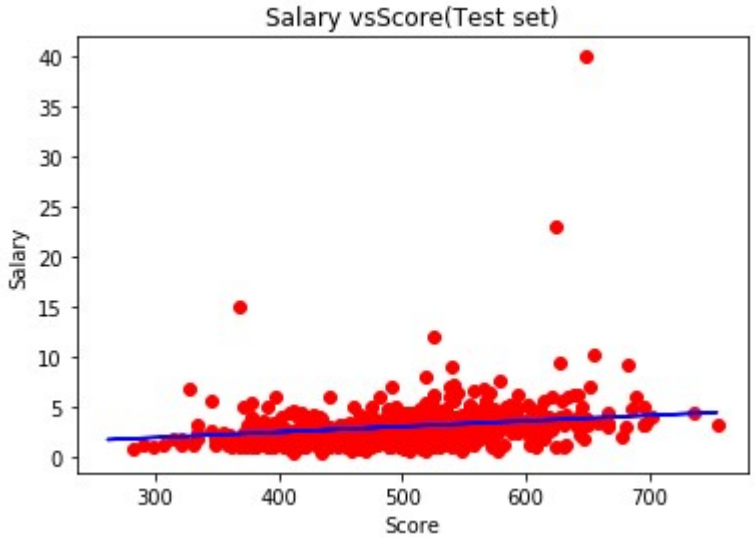
```
In [63]: import matplotlib.pyplot as plt
    ...: import pandas as pd
    ...:
    ...: df=pd.read_csv('G:\Data Analysis\output.csv')
    ...: df=df.dropna()
    ...:
    ...: df["normalised score"]=(df.English+ df.Logical+df.Quant)/3
    ...: X = df.iloc[:,[38]].values
    ...: y = df.iloc[:, 1].values

In [64]: from sklearn.model_selection import train_test_split
    ...: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 0)
    ...:
    ...:
    ...: from sklearn.linear_model import LinearRegression
    ...: regressor = LinearRegression()
    ...: regressor.fit(X_train, y_train)
Out[64]:
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)

In [65]: import statsmodels.api as sm
    ...:
    ...: model1=sm.OLS(y_train,X_train)
    ...: result=model1.fit()
    ...: print(result.summary())
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.690
Model:                            OLS   Adj. R-squared:                  0.690
Method:                 Least Squares   F-statistic:                     5805.
Date:                Sat, 06 Jul 2019   Prob (F-statistic):               0.00
Time:                        19:13:19   Log-Likelihood:                -5609.2
No. Observations:                2604   AIC:                         1.122e+04
Df Residuals:                    2603   BIC:                         1.123e+04
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.0061   7.99e-05     76.193      0.000       0.006       0.006
==============================================================================
Omnibus:                     3332.795   Durbin-Watson:                   2.007
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           795802.112
Skew:                           6.820   Prob(JB):                         0.00
Kurtosis:                      87.549   Cond. No.                         1.00
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [66]: c=0
    ...: y_pred = regressor.predict(X_test)
    ...: for i in range(len(y_pred)):
    ...:     y_pred[i]=(y_pred[i]<=y_test[i]+2  and y_pred[i]>=y_test[i]-2)
    ...:     if(y_pred[i]):
```

```
    ...:         c+=1
    ...:
    ...: acc=float(c/len(y_test))

In [67]: plt.scatter(X_train, y_train, color = 'red')
    ...: plt.plot(X_train, regressor.predict(X_train), color = 'blue')
    ...: plt.title('Salary vsScore(Training set)')
    ...: plt.xlabel('Score')
    ...: plt.ylabel('Salary')
    ...: plt.show()
    ...:
    ...:
    ...: plt.scatter(X_test, y_test, color = 'red')
    ...: plt.plot(X_train, regressor.predict(X_train), color = 'blue')
    ...: plt.title('Salary vsScore(Test set)')
    ...: plt.xlabel('Score')
    ...: plt.ylabel('Salary')
    ...: plt.show()
```




```
In [68]: acc
```

Out[68]: 0.8895705521472392

In [69]: