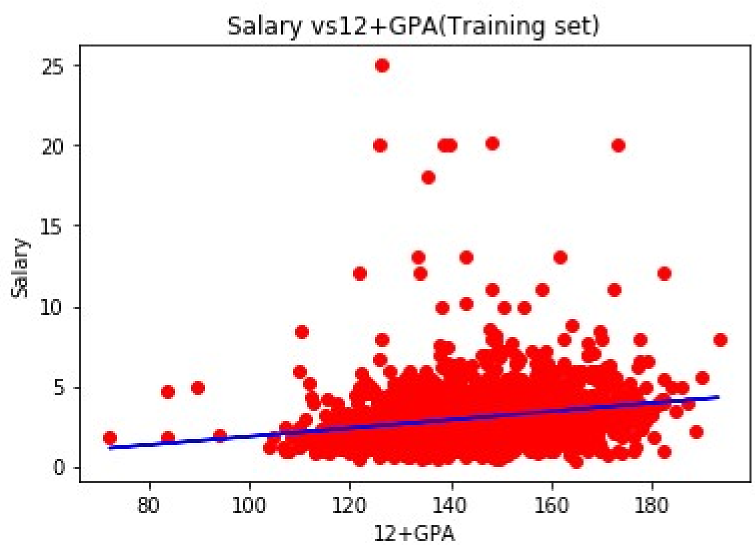
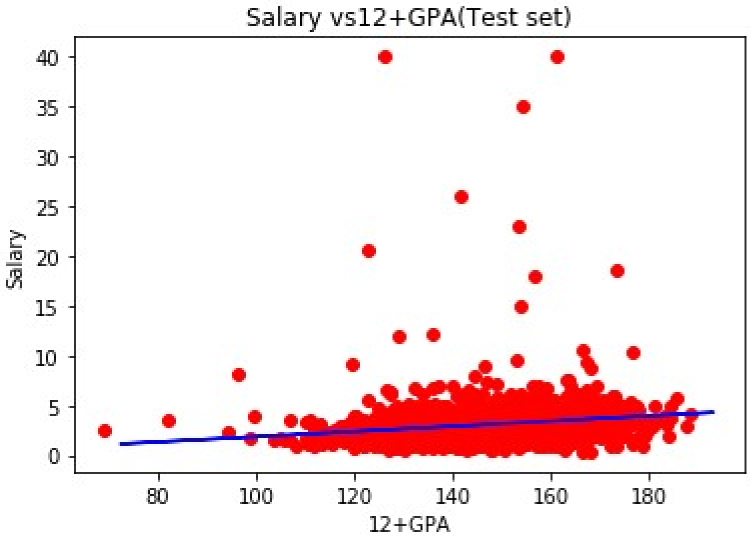
```
In [79]: import matplotlib.pyplot as plt
    ...: import pandas as pd
    ...:
    ...: df=pd.read_csv('G:\Data Analysis\output.csv')
    ...: df=df.dropna()
    ...:
    ...: df["normalised score"]=(df.t12percentage+ df.collegeGPA)
    ...: X = df.iloc[:,[38]].values
    ...: y = df.iloc[:, 1].values

In [80]: from sklearn.model_selection import train_test_split
    ...: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.4,
random_state = 0)
    ...:
    ...:
    ...: from sklearn.linear_model import LinearRegression
    ...: regressor = LinearRegression()
    ...: regressor.fit(X_train, y_train)
Out[80]:
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)

In [81]: 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:                       0.736
Model:                            OLS   Adj. R-squared:                  0.736
Method:                 Least Squares   F-statistic:                     5447.
Date:                Sat, 06 Jul 2019   Prob (F-statistic):               0.00
Time:                        19:22:41   Log-Likelihood:                -3959.6
No. Observations:                1953   AIC:                             7921.
Df Residuals:                    1952   BIC:                             7927.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.0210      0.000     73.806      0.000       0.020       0.022
==============================================================================
Omnibus:                     1781.729   Durbin-Watson:                   2.018
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            97469.145
Skew:                           4.121   Prob(JB):                         0.00
Kurtosis:                      36.613   Cond. No.                         1.00
==============================================================================

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

In [82]: 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 [83]: acc
Out[83]: 0.8810437452033768

In [84]: plt.scatter(X_train, y_train, color = 'red')
    ...: plt.plot(X_train, regressor.predict(X_train), color = 'blue')
    ...: plt.title('Salary vs12+GPA(Training set)')
    ...: plt.xlabel('12+GPA')
    ...: 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 vs12+GPA(Test set)')
    ...: plt.xlabel('12+GPA')
    ...: plt.ylabel('Salary')
    ...: plt.show()
```




Salary vs12+GPA(Test set)
Salary
40
35
30
25
20
15
10
5
0
80
100
120
140
160
180
12+GPA

In [85]: