

```

In [79]: import pandas as pd
...:
...: df=pd.read_csv('G:\Data Analysis\output.csv')
...: df=df.dropna()
...: X = df.iloc[:,[8,11,14,17,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37]].values
...: y = df.iloc[:, 1].values
...: for i in range(len(y)):
...:     y[i]=int(y[i])

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 = 0.3,
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]: c=0
...: y_pred = regressor.predict(X_test)
...: for i in range(len(y_pred)):
...:     y_pred[i]=int(y_pred[i])
...:     y_pred[i]=(y_pred[i]<=y_test[i]+2 and y_pred[i]>=y_test[i]-2)
...:     if(y_pred[i]):
...:         c+=1
...:
...: accuracy=float(c/len(y_test))

In [82]: accuracy
Out[82]: 0.9314227226202662

In [83]: import statsmodels.formula.api as sm
...: def backwardElimination(x, sl):
...:     numVars = len(x[0])
...:     for i in range(0, numVars):
...:         regressor_OLS = sm.OLS(y, x).fit()
...:         maxVar = max(regressor_OLS.pvalues).astype(float)
...:         if maxVar > sl:
...:             for j in range(0, numVars - i):
...:                 if (regressor_OLS.pvalues[j].astype(float) == maxVar):
...:                     x = np.delete(x, j, 1)
...:             regressor_OLS.summary()
...:         return x
...:
...:
...: SL = 0.05
...: import numpy as np
...: X=np.append(arr=np.ones((3256,1)).astype(int),values =X ,axis=1)
...: X_opt = X[:,[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]]
...: X_Modeled = backwardElimination(X_opt, SL)

In [84]: from sklearn.model_selection import train_test_split
...: X_train, X_test, y_train, y_test = train_test_split(X_Modeled, y, test_size = 0.3,
random_state = 0)

```

```

...:
...: from sklearn.linear_model import LinearRegression
...: regressor = LinearRegression()
...: regressor.fit(X_train, y_train)
Out[84]:
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
                normalize=False)

In [85]: c=0
...: y_pred = regressor.predict(X_test)
...: for i in range(len(y_pred)):
...:     y_pred[i]=int(y_pred[i])
...:     y_pred[i]=(y_pred[i]<=y_test[i]+2 and y_pred[i]>=y_test[i]-2)
...:     if(y_pred[i]):
...:         c+=1
...:
...: accuracy=float(c/len(y_test))

In [86]: accuracy
Out[86]: 0.9303991811668373

In [87]:

```