In [ ]:
```
# Author : Amir Shokri
# github link : https://github.com/amirshnll/Speaker-Accent-Recognition
# dataset link : http://archive.ics.uci.edu/ml/datasets/Speaker+Accent+Recogni
tion
# email : amirsh.nll@gmail.com
```

In [1]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import metrics
import seaborn as sn
import matplotlib.pyplot as plt

df = pd.read_csv('dataset.csv')

# x = df.drop('language', axis=1)
# y = df['language']


df = pd.DataFrame(df,columns= ['language','X1', 'X2','X3','X4', 'X5', 'X6', 'X
7', 'X8', 'X9', 'X10', 'X11', 'X12'])

# print(df)

x = df[['X1', 'X2','X3','X4', 'X5', 'X6', 'X7', 'X8', 'X9', 'X10', 'X11', 'X1
2']]
y = df['language']

x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_stat
e=0)

log_regression = LogisticRegression(max_iter=5000)
log_regression.fit(x_train, y_train)

y_pred = log_regression.predict(x_test)

# confusion_matrix = pd.crosstab(y_test, y_pred, rownames=['Actual'], colnames
=['Predicted'])
# sn.heatmap(confusion_matrix, annot=True)
print('========== Test Features ============')
print(x_test)

print('============ Predicted Values ===========')
print(y_pred)

print('Accuracy: ',metrics.accuracy_score(y_test, y_pred))

# plt.show()
```

```
========== Test Features ============
            X1         X2         X3         X4         X5         X6  \
122   4.915523  -1.235624   0.599622   7.392184  -6.940796  12.207071
66   -2.863910   1.608425  -1.575423   4.480974  -4.044910   9.854652
142  -0.604683  -6.917135   5.041063   9.041541  -6.151566   8.941550
246  -0.419643   1.996737   0.228772  -1.067420   2.047180   1.922359
146  -2.595753  -2.466705   0.420945   5.337807  -5.523456   9.905345
..         ...        ...        ...        ...        ...        ...
235   9.322440  -6.613469  15.545937  -0.692066   1.894247  -1.774044
74    5.260007  -4.809373   3.745731   4.557293  -5.853547  13.670204
52   11.010367  -7.384443   1.917493  14.137857  -7.562776   8.674802
236   6.027468  -4.932851  15.630876  -0.741649   5.298756   1.032338
215  10.262510 -14.294344   3.020109   8.522629 -10.627219  11.559337

            X7        X8        X9        X10        X11       X12
122  -9.472306  4.591876  1.369664  -4.563814   2.402545 -4.857425
66  -12.430810  7.228188 -1.795629  -1.801883   3.681854 -5.804928
142  -8.988871  6.992821  0.917500  -3.362391   0.311681 -7.100991
246  -5.937325  6.945288 -9.765117   7.552414  -3.122809 -1.647258
146 -10.124395  4.748502 -0.090000  -1.663290  -0.500254 -5.029416
..         ...       ...       ...        ...        ...       ...
235  -3.179173  4.184236 -7.829252  10.862655 -13.664104  3.627803
74  -10.862849  2.795047  1.535211  -1.147868   5.152335 -1.853668
52   -7.466133  6.909112  1.764772  -3.516485   8.485256 -7.841633
236  -4.786869  4.529260 -6.493262  10.275837 -11.121205  1.781395
215 -11.487250 -0.999699  1.695868  -2.700008   2.207667 -4.082932

[66 rows x 12 columns]
============ Predicted Values ===========
['US' 'IT' 'UK' 'US' 'UK' 'GE' 'US' 'ES' 'UK' 'UK' 'US' 'GE' 'UK' 'GE'
 'US' 'US' 'UK' 'US' 'GE' 'US' 'US' 'US' 'US' 'US' 'US' 'US' 'US' 'US'
 'FR' 'US' 'US' 'US' 'US' 'ES' 'ES' 'FR' 'US' 'ES' 'IT' 'GE' 'US' 'ES'
 'UK' 'US' 'ES' 'US' 'ES' 'US' 'IT' 'US' 'ES' 'UK' 'US' 'US' 'IT' 'ES'
 'US' 'GE' 'US' 'US' 'UK' 'US' 'US' 'FR' 'US' 'US']
Accuracy:  0.803030303030303
```