M.S.SANJAY

15BCE0517

L7+L8

EXPERIMENT 8

**SUPPORT VECTOR MACHINE**

Dataset:

**Glass Identification Data Set** (Multiclass dataset)

Attribute descriptions:

1. Id number: 1 to 214   
2. RI: refractive index   
3. Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10)   
4. Mg: Magnesium   
5. Al: Aluminum   
6. Si: Silicon   
7. K: Potassium   
8. Ca: Calcium   
9. Ba: Barium   
10. Fe: Iron   
11. Type of glass: (class attribute)   
-- 1 building\_windows\_float\_processed   
-- 2 building\_windows\_non\_float\_processed   
-- 3 vehicle\_windows\_float\_processed   
-- 4 vehicle\_windows\_non\_float\_processed **(none in this database)**   
-- 5 containers   
-- 6 tableware   
-- 7 headlamps

The last attribute is the target class and it takes more than two values(7 values) and thus it is multi class attribute.

**CODE**:

import numpy as np

import pandas as pd

from sklearn.preprocessing import StandardScaler

from sklearn.neural\_network import MLPClassifier

from sklearn import datasets, svm

from sklearn.metrics import confusion\_matrix

from sklearn.cross\_validation import train\_test\_split

from sklearn.metrics import accuracy\_score

from sklearn.metrics import classification\_report

balance\_data = pd.read\_csv('https://archive.ics.uci.edu/ml/machine-learning-databases/glass/glass.data',

sep= ',', header = 0 )

print ("Dataset Lenght: ", len(balance\_data))

print ("Dataset Shape: ", balance\_data.shape)

print ("Dataset: ",balance\_data.head())

X = balance\_data.values[:, 1:9]

Y = balance\_data.values[:, 10]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size = 0.3, random\_state = 100)

C = 1.0 # SVM regularization parameter

# SVC with linear kernel

svc = svm.SVC(kernel='linear', C=C).fit(X, Y)

predicted= svc.predict(X\_test)

cnf\_matrix = confusion\_matrix(y\_test, predicted)

print("Confusion matrix for SVC")

print(cnf\_matrix)

# SVC with RBF kernel

rbf\_svc = svm.SVC(kernel='rbf', gamma=0.7, C=C).fit(X, Y)

predicted= rbf\_svc.predict(X\_test)

cnf\_matrix = confusion\_matrix(y\_test, predicted)

print("Confusion matrix for RBF kernel")

print(cnf\_matrix)

# SVC with polynomial (degree 3) kernel

poly\_svc = svm.SVC(kernel='poly', degree=3, C=C).fit(X, Y)

predicted= poly\_svc.predict(X\_test)

cnf\_matrix = confusion\_matrix(y\_test, predicted)

print("Confusion matrix for POLYNOMIAL kernel")

print(cnf\_matrix)

***Output***:

Dataset Lenght: 213

Dataset Shape: (213, 11)

Dataset: 1 1.52101 13.64 4.49 1.10 71.78 0.06 8.75 0.00 0.00.1 1.1

0 2 1.51761 13.89 3.60 1.36 72.73 0.48 7.83 0.0 0.00 1

1 3 1.51618 13.53 3.55 1.54 72.99 0.39 7.78 0.0 0.00 1

2 4 1.51766 13.21 3.69 1.29 72.61 0.57 8.22 0.0 0.00 1

3 5 1.51742 13.27 3.62 1.24 73.08 0.55 8.07 0.0 0.00 1

4 6 1.51596 12.79 3.61 1.62 72.97 0.64 8.07 0.0 0.26 1

Confusion matrix for SVC

[[12 4 0 0 0 0]

[ 8 13 0 1 0 1]

[ 2 2 0 0 0 0]

[ 0 3 0 5 0 0]

[ 0 0 0 0 3 0]

[ 0 1 0 0 0 9]]

Confusion matrix for RBF kernel

[[14 2 0 0 0 0]

[ 3 20 0 0 0 0]

[ 2 2 0 0 0 0]

[ 0 0 0 8 0 0]

[ 0 1 0 0 2 0]

[ 1 0 0 0 0 9]]

Confusion matrix for POLYNOMIAL kernel

[[14 1 1 0 0 0]

[ 6 16 1 0 0 0]

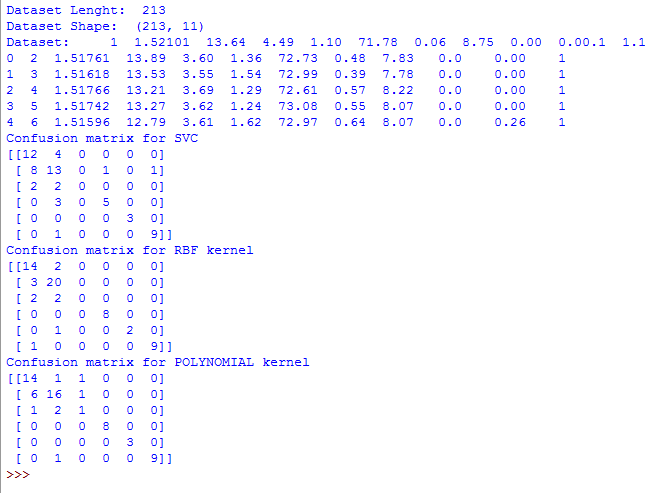
[ 1 2 1 0 0 0]

[ 0 0 0 8 0 0]

[ 0 0 0 0 3 0]

[ 0 1 0 0 0 9]]

**SCREENSHOT**:



EXPLANATION:

The confusion matrix is 6\*6 because as mentioned in the attribute information, the value 4 for “type of glass” attribute doesn’t occur in the dataset.