CLASS: B.E. E &TC **SUBJECT: ML**
EXPT. NO.: 7 **DATE:**
ROLL NO.: 42428

TITLE: Implement and test Multiclass SVM Classifier

CO 3: Design and implement machine learning solution to address specified problems of classification, regression, and clustering. Analyze the effect of dimensionality reduction using principal components analysis, factor analysis, multidimensional scaling. Evaluate and interpret the results of the algorithms.

CO 4: Carry out experiments as an individual and in a team, comprehend and write a laboratory record and draw conclusions at a technical level.

AIM:
To implement:
1. Multiclass Classification using SVM

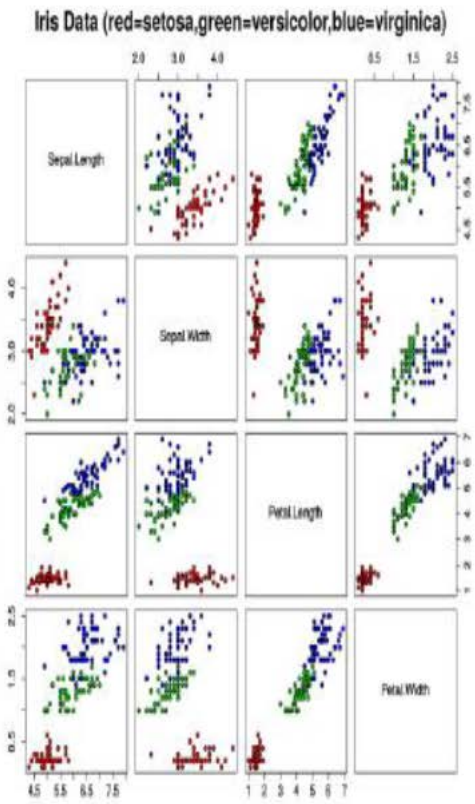
SOFTWARES REQUIRED: MATLAB 7.0 or Python

THEORY:

Fisher data for Iris flower is used for SVM Multi class data classification:

Dataset Order	Sepal length	Sepal width	Petal length	Petal width	Species
1	5.1	3.5	1.4	0.2	*I. setosa*
2	4.9	3.0	1.4	0.2	*I. setosa*
3	4.7	3.2	1.3	0.2	*I. setosa*
4	4.6	3.1	1.5	0.2	*I. setosa*
51	7.0	3.2	4.7	1.4	*I. versicolor*
52	6.4	3.2	4.5	1.5	*I. versicolor*
53	6.9	3.1	4.9	1.5	*I. versicolor*
148	6.5	3.0	5.2	2.0	*I. virginica*
149	6.2	3.4	5.4	2.3	*I. virginica*
150	5.9	3.0	5.1	1.8	*I. virginica*

Fisher Data



Compact Classification of ECOC Class:

Compact multiclass model for support vector machines or other classifiers.

Description:

Compact Classification ECOC is a compact, error-correcting output codes (ECOC) multiclass model.

The compact classifier does not include the data used for training the ECOC multiclass model. Therefore, you cannot perform tasks, such as cross validation, using the compact classifier. Use a compact ECOC multiclass model for labeling new data (in other words, predicting the labels of new data).

ALGORITHM:

1. Load Fisher's iris data set for Multiclass Data, typically Iris flower 3 class data set.
2. Remove empty classes.
3. Display class Names which are the unique classes in the data set and K is the number of classes.
4. For reproducibility, seeds the random number generator using the nonnegative integer seed so that rand, randi, and randn produce a predictable sequence of numbers.
5. For a one-versus-all coding design, there are K = 3 binary learners. Specify templates for the binary learners such that:
   - Binary learner 1 and 2 are naive Bayes classifiers. By default, each predictor is conditionally, normally distributed given its label.
   - Binary learner 3 is an SVM classifier. Specify to use the Gaussian kernel.
6. For that use, tLearners = {tNB tNB tSVM};
   where tNB and tSVM are template objects for naive Bayes and SVM learning, respectively.
7. Fit multiclass models for support vector machines or other classifiers using CVMdl. CVMdl is a Classification Partitioned ECOC(Error Correcting Output Codes) cross- validated model.
8. Inspect one of the trained folds using dot notation such as CVMdl.Trained{1} Each fold is a Compact Classification ECOC model trained on 90% of the data.
9. Display the trained SVM classifier (the third binary learner) in the first fold such as : CVMdl.Trained{1}.BinaryLearners{3}
10. Estimate the generalization error

CONCLUSION:

The proposed support vector Machine in Binary Tree Architecture method was designed to provide superior recognition speed utilizing decision tree architecture, while keeping comparable recognition rate to the other known methods. Clustering algorithm that utilizes distance measures at the kernel space is used to convert the multiclass problem into binary tree, in which the binary tree measures is made by the SVMs. SVM-BTA is becoming more

favorable to other compared methods as the number of classes in the problem increases.

REFERENCES:

i. Laurene Fausett ,"Fundamentals of Neural Networks: Architectures, Algorithms and Applications", Pearson Education, Inc, 2008.

ii. S. N. Sivanandam , S. Sumathi, S. N. Deepa, "Introduction to Neural Networks using MATLAB", McGraw Hill, 2006.

iii. S. N. Sivanandam, S. N. Deepa, "Principles of Soft Computing" , John Wiley & Sons, 2007

iv. Phil Kim, "MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence", a Press 2017.

(Course Teacher)

Code :

```
import numpy as np
import cvxopt
from sklearn import datasets
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from matplotlib import pyplot as plt
from sklearn.svm import SVC

iris = datasets.load_iris()
# Take the first two features. We could avoid this by using a two-dim dataset
x = iris.data
y = iris.target

x_train,x_test,y_train,y_test = train_test_split(x,y,random_state=0)

#training a linear SVM model
svm_linear_model = SVC(kernel='linear', C = 1.0).fit(x_train,y_train)
svm_predictions = svm_linear_model.predict(x_test)
print(svm_predictions)

#model accuracy for x_test
accuracy = svm_linear_model.score(x_test, y_test)
print(accuracy)

#creating confusion matrix
cm = confusion_matrix(y_test, svm_predictions)
print(cm)
```

Output :

```
[2 1 0 2 0 2 0 1 1 1 2 1 1 1 1 0 1 1 0 0 2 1 0 0 2 0 0 1 1 0 2 1 0 2 2 1 0
 2]
0.9736842105263158
[[13  0  0]
 [ 0 15  1]
 [ 0  0  9]]
PS D:\Study\Practical Work\4th year\ML\24-05-2021>
```