

## LAB EXERCISE

### SUPPORT VECTOR MACHINES

#### Aim of the Experiment

The main aim of this experiment is to implement support vector machine for the Iris dataset.

The objectives of this experiment are

1. Implement SVM for Iris Dataset
2. Find confusion matrix and evaluation metrics for SVM

SVM model can be constructed using sklearn command,

```
model = SVC(kernel='linear',random_state=0)
```

Similarly, by changing the flag 'linear' with 'rbf', one can construct the Gaussian RBF kernel also.

model.fit and model.predict can be used to fit the data and to make prediction.

#### Program Listing - 1

```
import pandas as pd

from sklearn.svm import SVC

from sklearn.model_selection import train_test_split

from sklearn.metrics import confusion_matrix

from sklearn.metrics import classification_report

from sklearn.metrics import accuracy_score


# Reading the csv Iris dataset file

df = pd.read_csv("iris1.csv")

print(df.head(10))

# Conditioning the data

array = df.values

X = array[:,0:4]

y = array[:,4]

# Condition the training and Testing data

# The number of samples can be tuned with the test_size parameter.

# Here, 95% of the data is used.
```

```

X_train, X_test, y_train, y_test = train_test_split( \
    X, y, test_size=0.95, random_state=0)
# Construct the Linear model
model = SVC(kernel='linear', random_state=0)
model.fit(X_train, y_train)
# Test the model with Linear kernel
pred = model.predict(X_test)
# Prepare confusion matrix
print("\n\nThe confusion matrix is \n\n")
conf = confusion_matrix(y_test, pred)
print(conf)
# prepare Classification Report
print("\n\nAccuracy is")
accuracy = accuracy_score(y_test, pred)
print(accuracy)
# Or report can be obtained as follows
print('\n Classification Report')
report = classification_report(y_test, pred)
print(report)
# RBF kernel
modell = SVC(kernel='rbf', random_state=0)
modell.fit(X_train, y_train)
# Test the model
pred = modell.predict(X_test)
# Prepare confusion matrix
print("\n\nThe confusion matrix for RBF kernel is \n\n")
conf = confusion_matrix(y_test, pred)
print(conf)

```

```
# prepare Classification Report
print("\n\nAccuracy for RBF kernel is")
accuracy = accuracy_score(y_test,pred)
print(accuracy)
```

### Output

The following screenshot shows the first ten records of Iris Dataset.

```
5.1 3.5 1.4 0.2 Setosa
0 5.1 3.5 1.4 0.2 Setosa
1 4.9 3.0 1.4 0.2 Setosa
2 4.7 3.2 1.3 0.2 Setosa
3 4.6 3.1 1.5 0.2 Setosa
4 5.0 3.6 1.4 0.2 Setosa
5 5.4 3.9 1.7 0.4 Setosa
6 4.6 3.4 1.4 0.3 Setosa
7 5.0 3.4 1.5 0.2 Setosa
8 4.4 2.9 1.4 0.2 Setosa
9 4.9 3.1 1.5 0.1 Setosa
```

The following screenshot shows the confusion matrix of SVM with linear kernel.

```
The confusion matrix is

[[46  0  0]
 [ 0 40  9]
 [ 0  1 47]]

Accuracy is
0.9300699300699301
```

The following screenshot shows the classification report of SVM with linear kernel.

```
Classification Report
precision    recall  f1-score   support

   Setosa   1.00    1.00    1.00     46
Versicolor  0.98    0.82    0.89     49
  Virginica  0.84    0.98    0.90     48

 accuracy   0.93
macro avg   0.94    0.93    0.93    143
weighted avg 0.94    0.93    0.93    143
```

The following screenshot shows the confusion matrix of SVM with Gaussian RBF kernel.

```
The confusion matrix for RBF kernel is
```

```
[[46  0  0]
 [ 7  0 42]
 [ 0  0 48]]
```

```
Accuracy for RBF kernel is
0.6573426573426573
```