G H Raisoni College of Engineering & Management, Pune

NAAC Accredited A+ Grade

(An Empowered Autonomous Institute Affiliated to Savitribai Phule Pune University)

Department of CSE Artificial Intelligence

Experiment No. 7

Title: Implement Support Vector Machine algorithm.

Objectives:

1. Implement the Support Vector Machine (SVM) algorithm for classification.
2. Understand hyperplane separation and kernel trick for non-linear classification.
3. Evaluate SVM using accuracy.

Problem Statement:

Develop a Support Vector Machine (SVM) model to classify a given dataset into different categories. The dataset may contain non-linearly separable data, requiring the use of kernel functions. The goal is to find the best hyperplane that maximizes the margin between different classes while minimizing misclassification.

Outcomes:

1. Successfully implement SVM for classification tasks.
2. Identify optimal hyperplane for given data.
3. Compare different kernel functions (Linear, Polynomial, RBF).
4. Evaluate model performance using confusion matrix and accuracy metrics.

Tools Required: 4GB RAM, Anaconda, Notebook

Theory:

Hyperplane: A decision boundary separating different classes.
Support Vectors: Data points closest to the hyperplane, influencing its position.
Margin: Distance between the hyperplane and support vectors (maximize it).
Kernel Trick: Transforming non-linearly separable data into higher-dimensional space.
Types of SVM:

- Linear SVM (when data is linearly separable).
- Non-linear SVM (using kernel functions: Polynomial, Radial Basis Function (RBF)).

Algorithm:

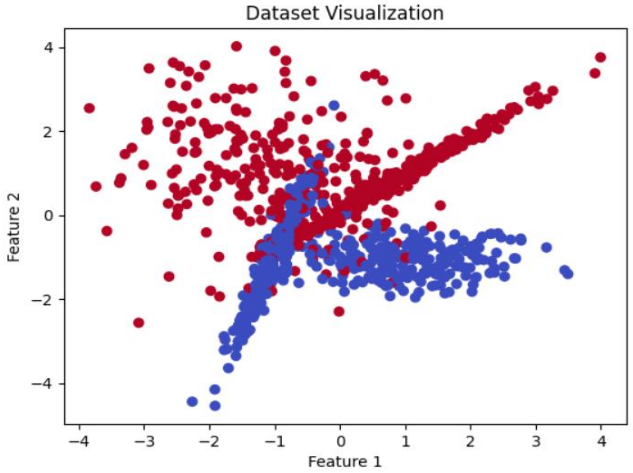
Steps:

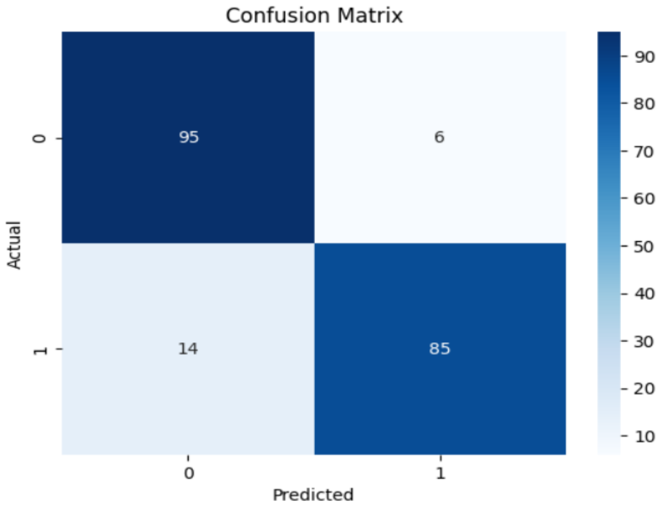
1. Input: Dataset with features and labels.
2. Preprocessing: Normalize/scale the data.
3. Train-Test Split: Divide dataset into training and testing sets.
4. Choose Kernel Function: Linear, Polynomial, or RBF.
5. Train SVM Model: Use Scikit-learn's SVM classifier.
6. Predict: Classify test data.
7. Evaluate: Compute accuracy, precision, recall, F1-score.
8. Optimize (Optional): Tune hyperparameters using GridSearchCV.

Source Code:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.datasets import make_classification
X, y = make_classification(n_samples=1000,
                           n_features=2,
                           n_informative=2,
                           n_redundant=0,
                           n_repeated=0,
                           n_classes=2,
                           random_state=42)
plt.scatter(X[:, 0], X[:, 1], c=y, cmap='coolwarm')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.title('Dataset Visualization')
plt.show()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
svm_model = SVC(kernel='rbf', C=1.0, gamma='scale')
svm_model.fit(X_train, y_train)
y_pred = svm_model.predict(X_test)
conf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
print(classification_report(y_test, y_pred))
```

Output:





```
                 precision    recall    f1-score   support
            0       0.87          0.94      0.90        101
            1       0.93          0.86      0.89        99
accuracy                                   0.90        200
macro avg         0.90          0.90      0.90        200
weighted avg    0.90          0.90      0.90        200
```

Conclusion:

____

____

____

____