Assignment – 7

1. PERFORM SUPPORT VECTOR MACHINE ON CLASSIFICATION.

Aim:

To implement **Support Vector Machine (SVM) for classification**, train the model on a dataset, and visualize the **decision boundary** to analyze its performance.

Algorithm:

1. Import Required Libraries → Use pandas, sklearn.svm, train_test_split, and accuracy_score.
2. Load a Dataset → Example: Use sklearn.datasets.load_iris() or any classification dataset.
3. Split Data into Training & Testing Sets → Use train_test_split().
4. Train SVM Model
   - Use SVC(kernel='linear') for a linear classifier.
   - Use SVC(kernel='rbf') for a non-linear classifier.
5. Make Predictions → Use the trained model to predict test data.
6. Evaluate Performance → Calculate accuracy using accuracy_score().

CODE:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

iris = datasets.load_iris()
X = iris.data[:, :2]
y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = SVC(kernel='linear')
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("SVM Classification Accuracy:", accuracy)



def plot_decision_boundary(model, X, y):
    x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
    y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
```

```
    xx, yy = np.meshgrid(np.linspace(x_min, x_max, 100), np.linspace(y_min, y_max, 100))

    Z = model.predict(np.c_[xx.ravel(), yy.ravel()])
    Z = Z.reshape(xx.shape)

    plt.contourf(xx, yy, Z, alpha=0.3)
    plt.scatter(X[:, 0], X[:, 1], c=y, edgecolors='k', marker='o')
    plt.title("SVM Classification Decision Boundary")
    plt.xlabel("Feature 1")
    plt.ylabel("Feature 2")
    plt.show()
plot_decision_boundary(model, X, y)
```

Output :