SVKM's NMIMS

Mukesh Patel School of Technology Management & Engineering

Program: B Tech/ MBA Tech Artificial Intelligence, B Tech (AI and ML, AI and DS, CSE (DS))

Course: Machine Learning

Experiment No.08

PART B

Roll No.: E038	Name: Arushi Pandey
Class : 2nd year STME	Batch : 2023-27
Date of Experiment: 2-04-25	Date of Submission: 8-04-25

B.1 Tasks

Task 1:

1. Implement SVM classifier on data generated.
2. Generate data using make_blobs (50 samples, two features per sample).
3. Plot a scatter plot of the samples.
4. Create linear SVM model using Sklearn.
5. Plot the decision boundary.
6. Determine the support vectors.
7. Vary the regularization parameter and tune the model.

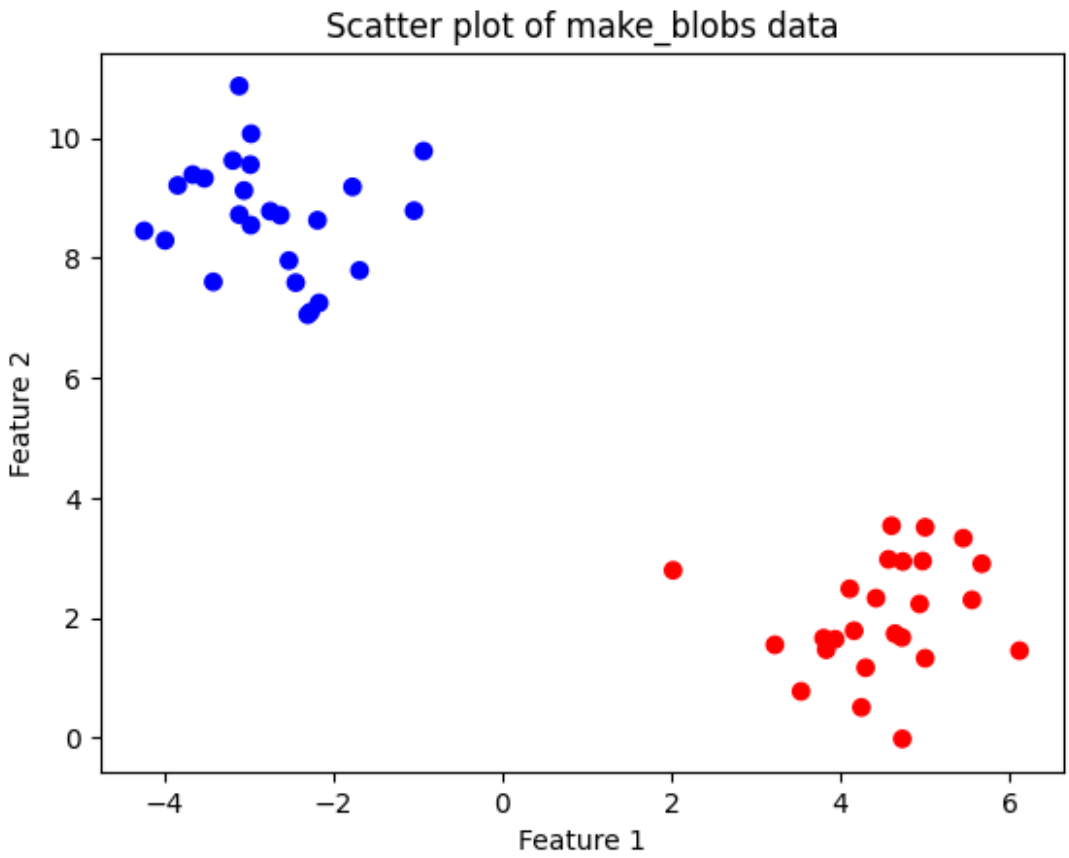
Task 2:

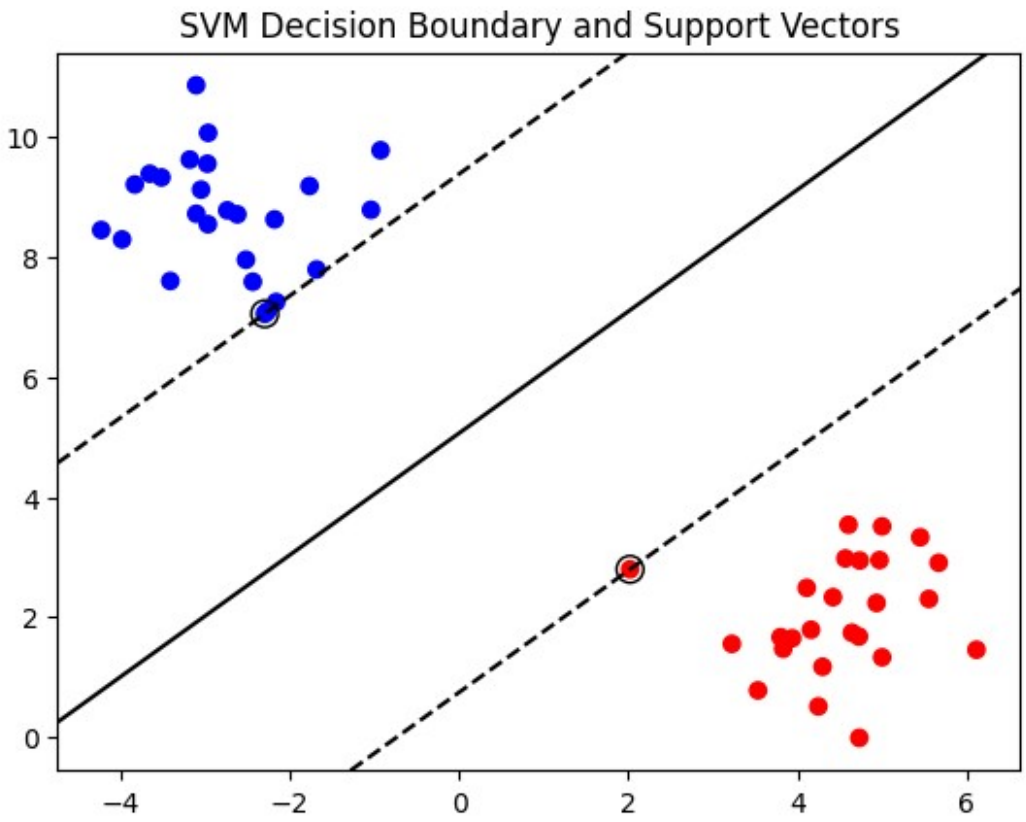
1. Import LFW dataset from sklearn.datasets.
2. Display a sample of the dataset.
3. Use SVM for classification.
4. Generate classification report.

B.4 Conclusion:

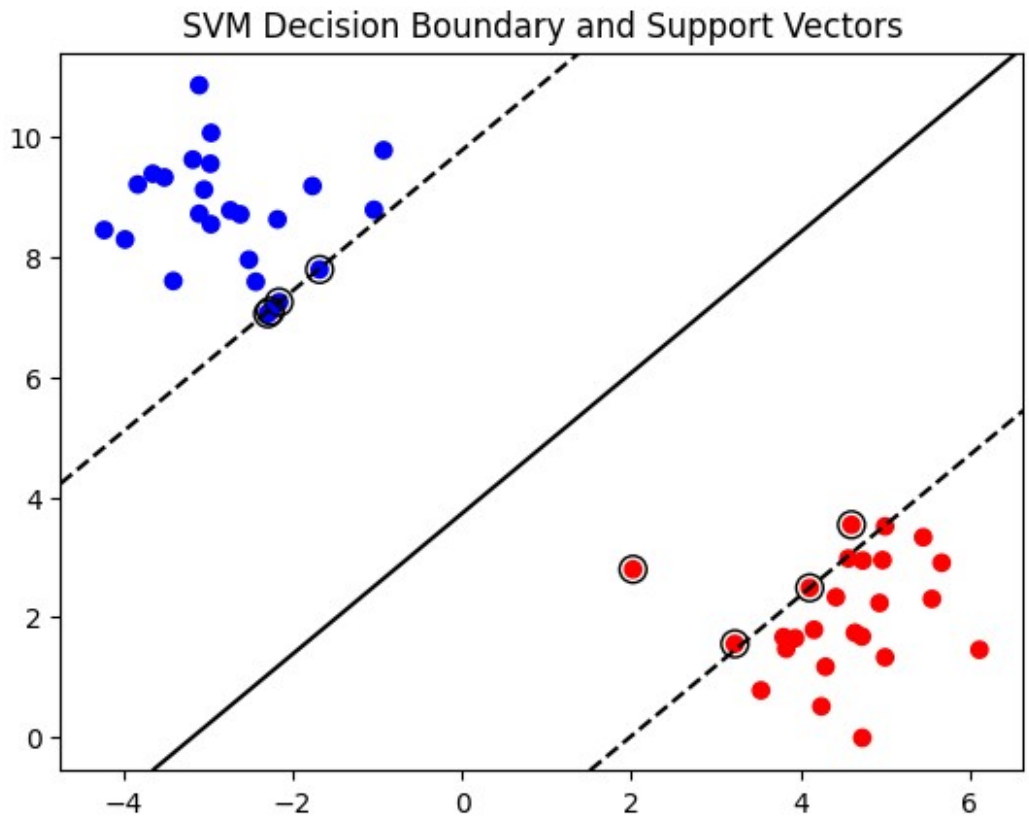
creating a robust decision In this experiment, we successfully implemented the Support Vector Machine (SVM) algorithm for classification tasks. SVM proved to be a powerful supervised learning technique capable of handling linear as well as non-linear classification problems using kernel functions. By experimenting with various kernels like linear, polynomial, and RBF, we were able to observe changes in model performance. Overall, the experiment demonstrated how SVM can effectively classify complex datasets by focusing on the most influential data points—support vectors—thereby boundary.

```
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.svm import SVC
import numpy as np

X, y = make_blobs(n_samples=50, centers=2, n_features=2,
random_state=42)

plt.scatter(X[:, 0], X[:, 1], c=y, cmap='bwr')
plt.title("Scatter plot of make_blobs data")
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.show()

clf = SVC(kernel='linear', C=1.0)
clf.fit(X, y)

def plot_decision_boundary(clf, X, y):
    plt.scatter(X[:, 0], X[:, 1], c=y, cmap='bwr')
    ax = plt.gca()

    xlim = ax.get_xlim()
    ylim = ax.get_ylim()

    xx = np.linspace(xlim[0], xlim[1], 30)
    yy = np.linspace(ylim[0], ylim[1], 30)
    YY, XX = np.meshgrid(yy, xx)
    xy = np.vstack([XX.ravel(), YY.ravel()]).T
    Z = clf.decision_function(xy).reshape(XX.shape)
    ax.contour(XX, YY, Z, colors='k', levels=[-1, 0, 1],
linestyles=['--', '-', '--'])

    ax.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1],
s=100,
               linewidth=1, facecolors='none', edgecolors='k')
    plt.title("SVM Decision Boundary and Support Vectors")
    plt.show()

plot_decision_boundary(clf, X, y)

for C in [0.01, 0.1, 1, 10]:
    clf = SVC(kernel='linear', C=C)
    clf.fit(X, y)
    print(f"\nRegularization Parameter C = {C}")
    plot_decision_boundary(clf, X, y)
```

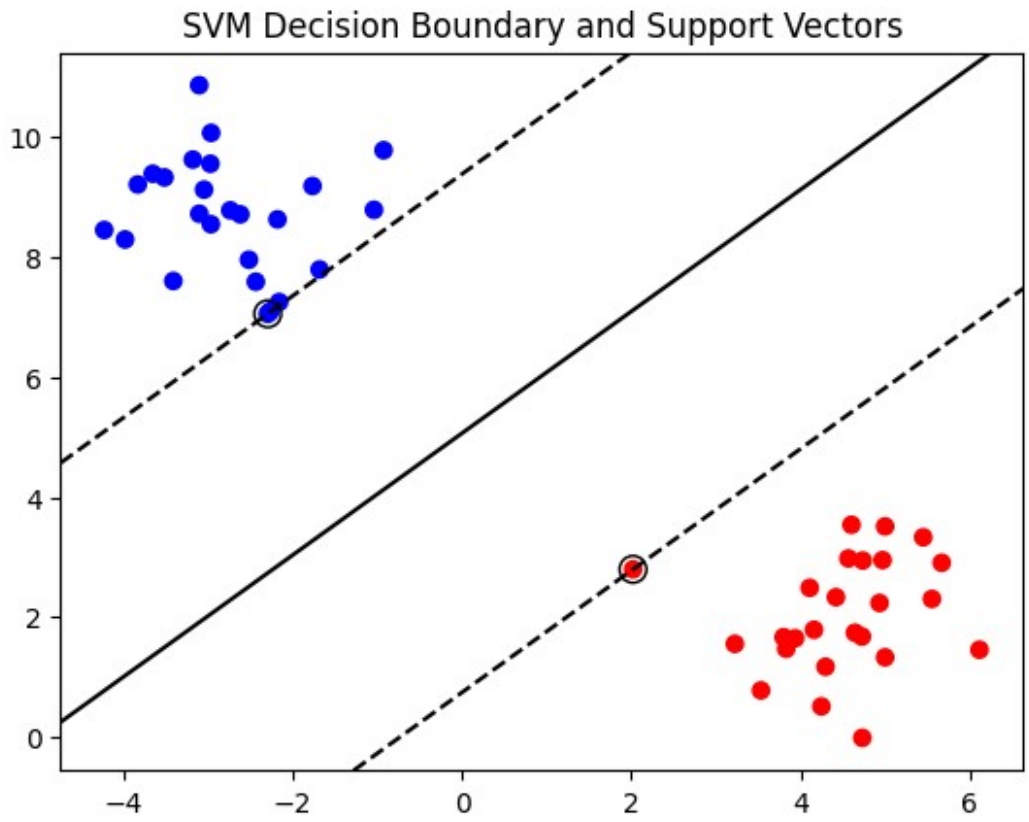

Scatter plot of make_blobs data
Feature 2
Feature 1
10
8
6
4
2
0
−4
−2
0
2
4
6



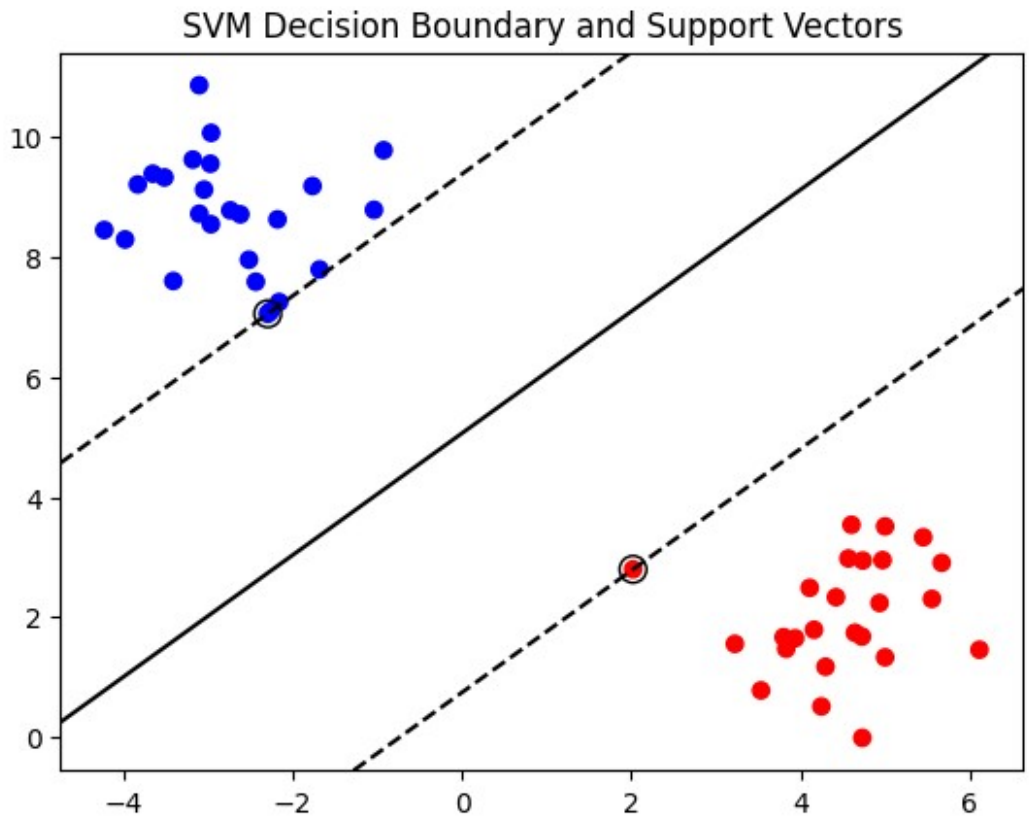

```
Regularization Parameter C = 0.01
```



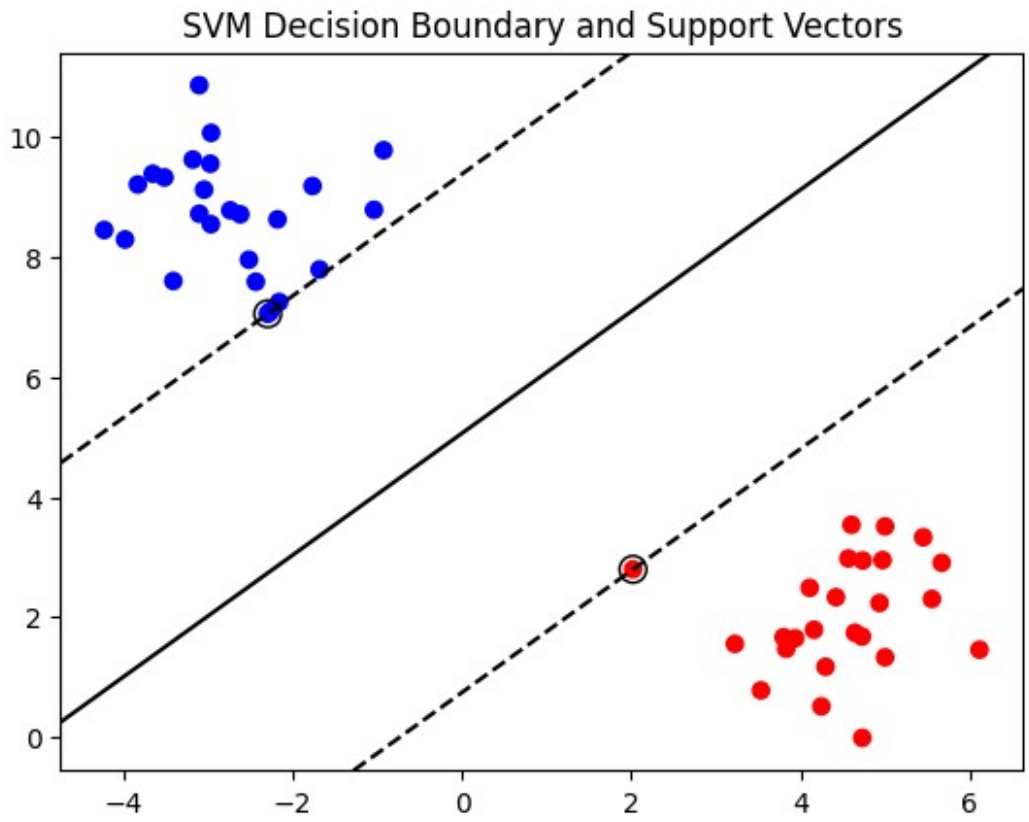

```
Regularization Parameter C = 0.1
```




```
Regularization Parameter C = 1
```



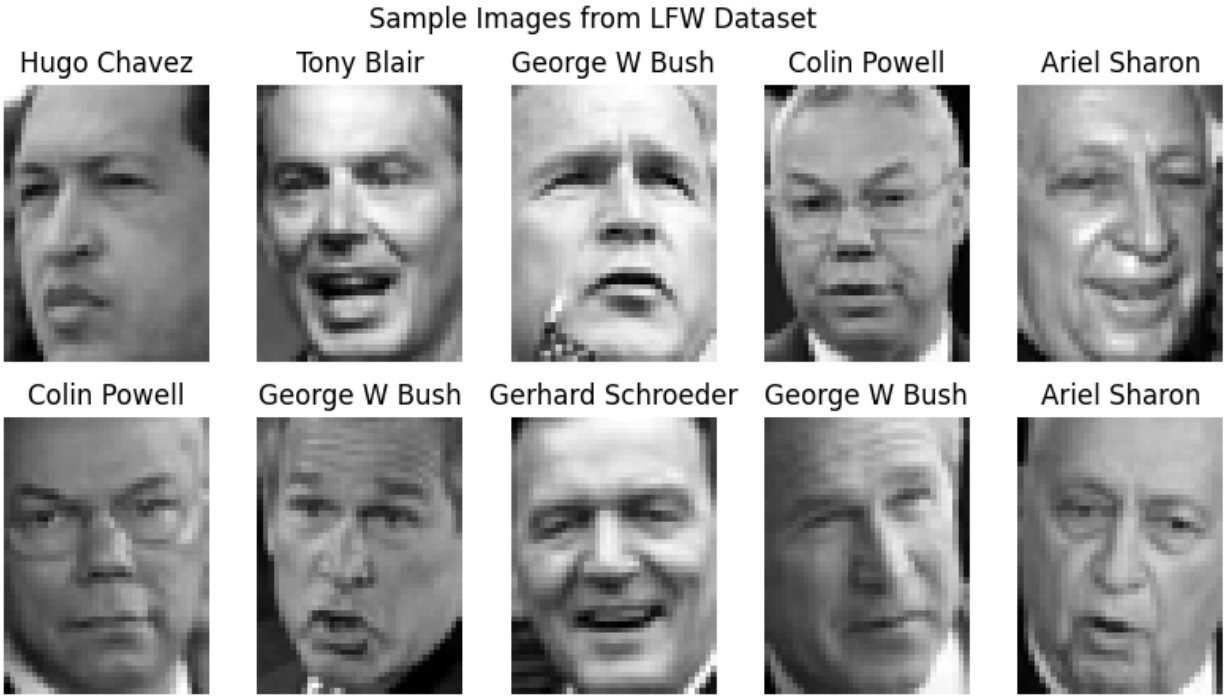

```
Regularization Parameter C = 10
```




```
from sklearn.datasets import fetch_lfw_people
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report
import matplotlib.pyplot as plt

lfw = fetch_lfw_people(min_faces_per_person=70, resize=0.4)
X = lfw.data
y = lfw.target
target_names = lfw.target_names

print("Dataset shape:", X.shape)
print("Number of classes:", len(target_names))

fig, axes = plt.subplots(2, 5, figsize=(10, 5))
for i, ax in enumerate(axes.flat):
    ax.imshow(lfw.images[i], cmap='gray')
    ax.set_title(target_names[lfw.target[i]])
    ax.axis('off')
plt.suptitle("Sample Images from LFW Dataset")
plt.show()

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

```
clf = SVC(kernel='linear', class_weight='balanced')
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
print("\nClassification Report:\n")
print(classification_report(y_test, y_pred,
target_names=target_names))
```

```
Dataset shape: (1288, 1850)
Number of classes: 7
```




```
Classification Report:

                   precision    recall  f1-score   support

     Ariel Sharon       0.67      0.73      0.70        11
     Colin Powell       0.77      0.91      0.83        47
  Donald Rumsfeld       0.68      0.77      0.72        22
    George W Bush       0.95      0.88      0.92       119
Gerhard Schroeder       0.78      0.95      0.86        19
      Hugo Chavez       1.00      0.69      0.82        13
       Tony Blair       0.87      0.74      0.80        27

         accuracy                           0.85       258
        macro avg       0.82      0.81      0.81       258
     weighted avg       0.87      0.85      0.85       258
```