lab-12-hajira-imran-44594

April 29, 2025

[1]:
```
# Step 1: Upload the CSV
from google.colab import files
uploaded = files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving svm_dataset.csv to svm_dataset (1).csv
```

[6]:
```
import pandas as pd

# Step 2: Load and preprocess the data
df = pd.read_csv(next(iter(uploaded)))

# Print column names
print("Columns in your CSV:")
print(df.columns)
```

```
Columns in your CSV:
Index(['ID', 'Feature1 (X1)', 'Feature2 (X2)', 'Label (Y)'], dtype='object')
```

[7]:
```
X = df[['Feature1 (X1)', 'Feature2 (X2)']]
y = df['Label (Y)']
```

[8]:
```
# Step 4: Train/test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
↪random_state=42)
```

[9]:
```
# Step 5: Train the SVM model (linear kernel)
from sklearn.svm import SVC
model = SVC(kernel='linear')
model.fit(X_train, y_train)
```

[9]:
```
SVC(kernel='linear')
```

[10]:
```
# Step 6: Evaluate model accuracy
from sklearn.metrics import accuracy_score
y_pred = model.predict(X_test)
```

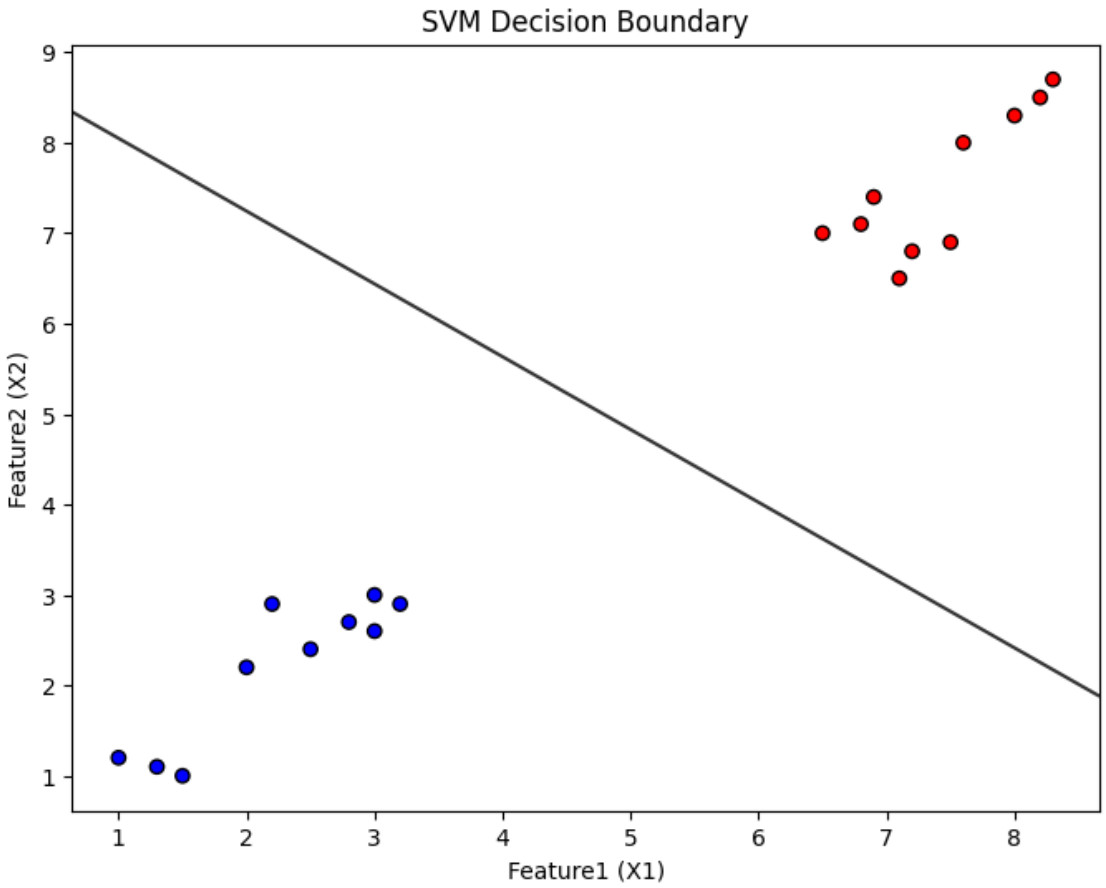
```
accuracy = accuracy_score(y_test, y_pred)
print(f"\nAccuracy on test set: {accuracy:.2f}")
```

```
Accuracy on test set: 1.00
```

[16]:
```
import matplotlib.pyplot as plt
import numpy as np

def plot_svm_boundary(model, X, y):
    plt.figure(figsize=(8, 6))

    # Scatter plot of data points
    plt.scatter(X['Feature1 (X1)'], X['Feature2 (X2)'], c=y, cmap='bwr',␣
 ↪edgecolors='k')

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

    # Create grid to evaluate the model
    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 = model.decision_function(xy).reshape(XX.shape)

    # Plot decision boundary
    ax.contour(XX, YY, Z, colors='k', levels=[0], alpha=0.8, linestyles=['-'])

    plt.xlabel('Feature1 (X1)')
    plt.ylabel('Feature2 (X2)')
    plt.title('SVM Decision Boundary')
    plt.show()
```

[17]:
```
# Step 7: Call the function to display the plot
plot_svm_boundary(model, X, y)
```

```
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739:
UserWarning: X does not have valid feature names, but SVC was fitted with
feature names
  warnings.warn(
```


SVM Decision Boundary
Feature2 (X2)
Feature1 (X1)
1
2
3
4
5
6
7
8
9