lab12-ai-alishba-waqar-46997

April 29, 2025

[1]:
```
from google.colab import files
uploaded = files.upload()
```

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

```
Saving svm_dataset.csv to svm_dataset.csv
```

[11]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
from mlxtend.plotting import plot_decision_regions
```

[12]:
```
df = pd.read_csv("svm_dataset.csv")
print(df.columns)
```

```
Index(['ID', 'Feature1 (X1)', 'Feature2 (X2)', 'Label (Y)'], dtype='object')
```

[13]:
```
X = df[["Feature1 (X1)", "Feature2 (X2)"]].values
y = df["Label (Y)"].values
```

[14]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,␣
↪random_state=42)
```

[15]:
```
model = SVC(kernel='linear')
model.fit(X_train, y_train)
```

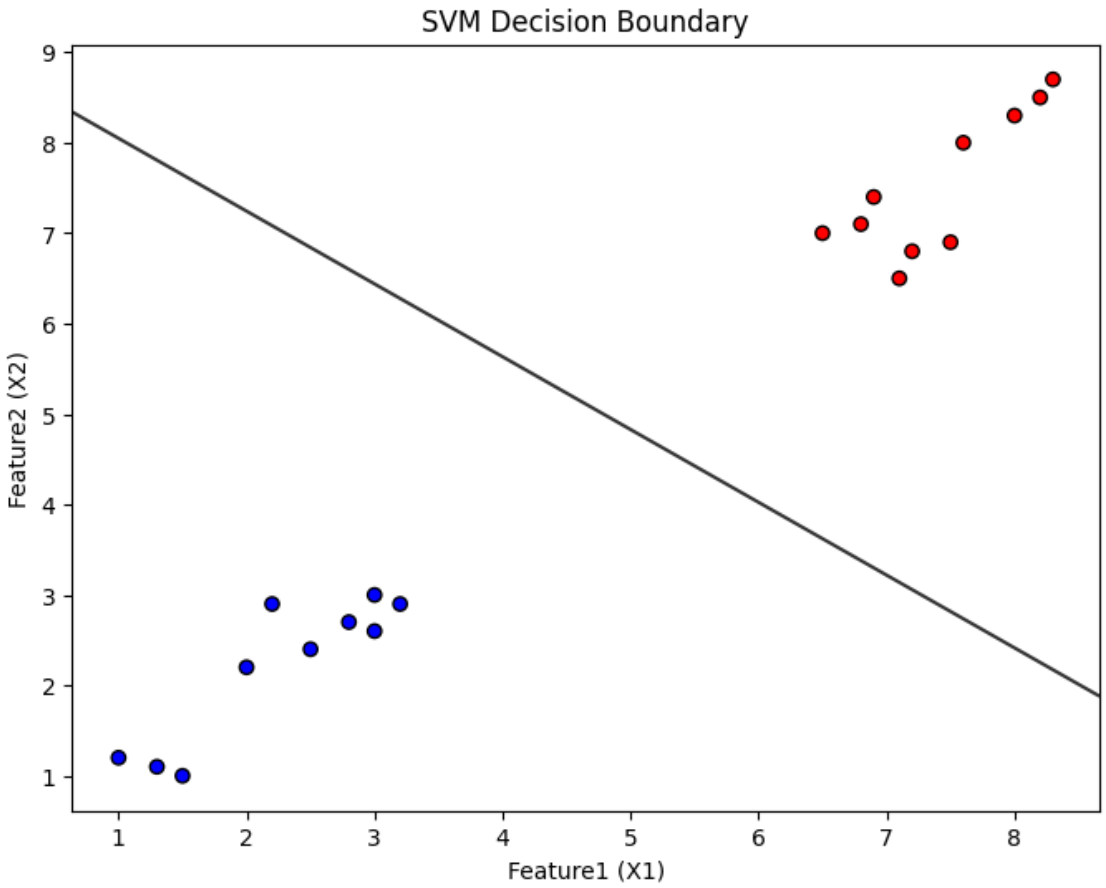
[15]:
```
SVC(kernel='linear')
```

[16]:
```
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy on test set: {accuracy:.2f}%")
```

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

[23]:
```
def plot_svm_boundary(model, X, y):
    plt.figure(figsize=(8, 6))

    # Scatter plot of data points
    # Accessing data in the NumPy array using indexing
    plt.scatter(X[:, 0], X[:, 1], c=y, cmap='bwr', edgecolors='k')
    # X[:, 0] selects all rows (:) and the first column (0) - Feature1 (X1)
    # X[:, 1] selects all rows (:) and the second column (1) - Feature2 (X2)


    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()
```

[24]:
```
#Call the function to display the plot
plot_svm_boundary(model, X, y)
```


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