```
# Import libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('svm_dataset.csv')

data.columns
```

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

```
X = data[['Feature1 (X1)', 'Feature2 (X2)']].values
y = data['Label (Y)'].values

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=42)

# Train model
model = SVC(kernel='linear')
model.fit(X_train, y_train)
```

```
SVC(kernel='linear')
```

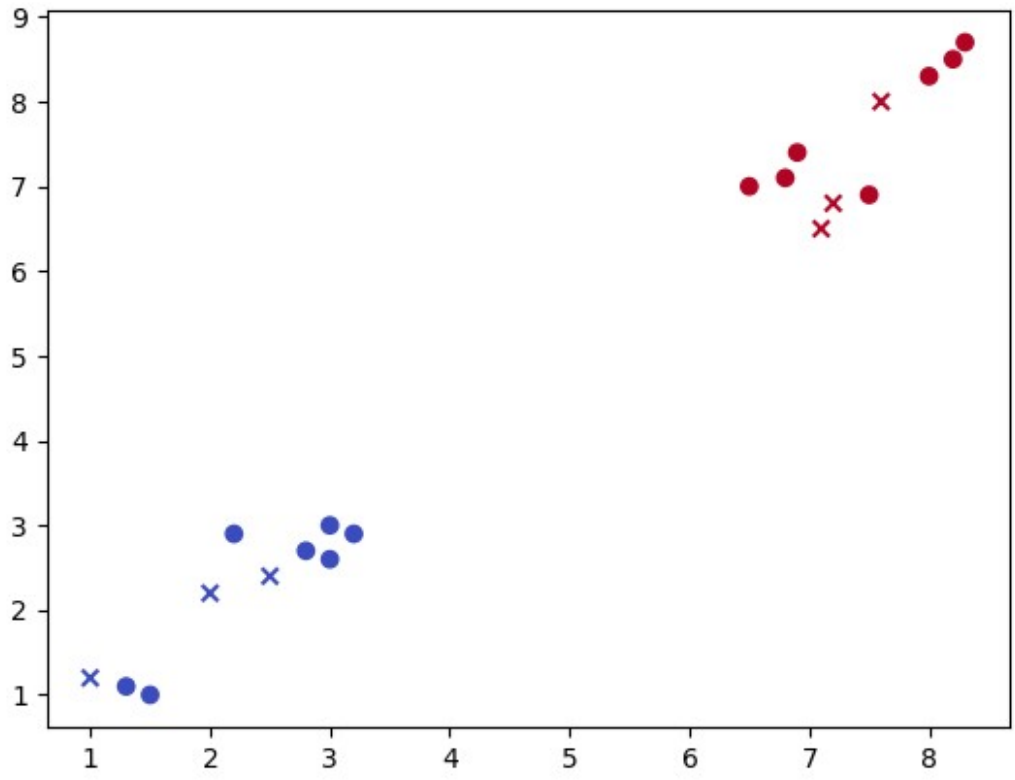
```
# Accuracy
print("Accuracy:", model.score(X_test, y_test))
```

```
Accuracy: 1.0
```

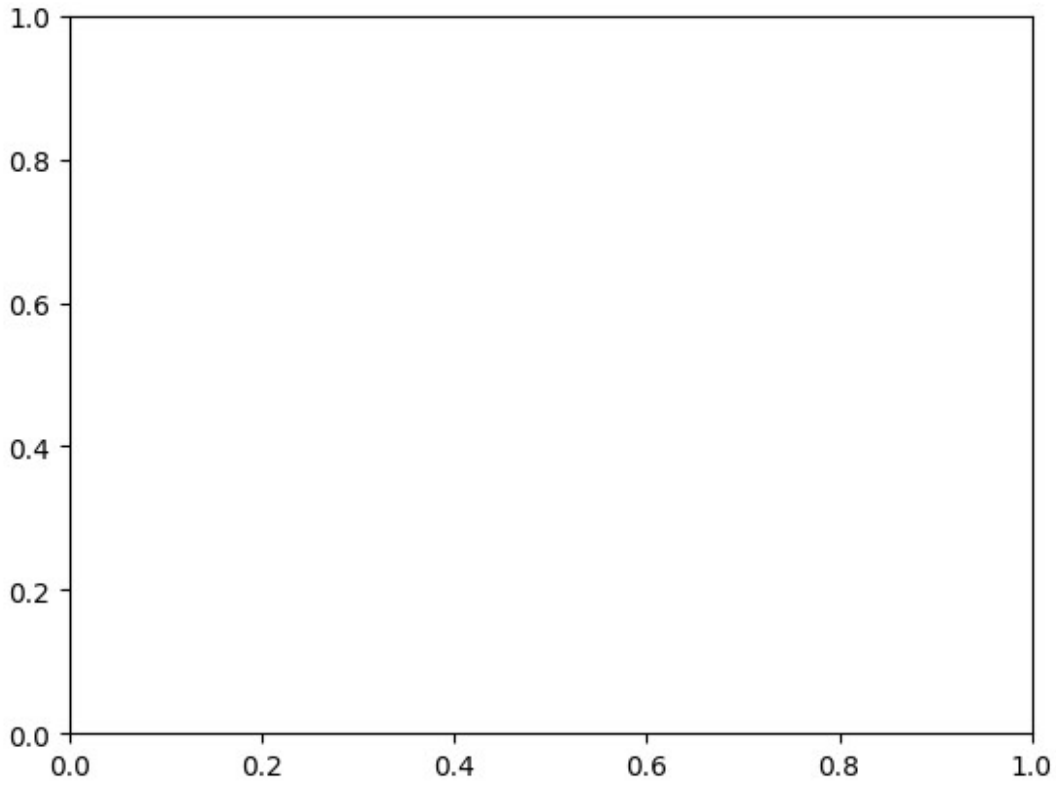
```
# Plotting train and test data separately
plt.scatter(X_train[:, 0], X_train[:, 1], c=y_train, cmap='coolwarm',
label='Train')
plt.scatter(X_test[:, 0], X_test[:, 1], c=y_test, cmap='coolwarm',
edgecolors='k', marker='x', label='Test')
```

```
<ipython-input-10-bbc22ab9b228>:3: UserWarning: You passed a
edgecolor/edgecolors ('k') for an unfilled marker ('x').  Matplotlib
is ignoring the edgecolor in favor of the facecolor.  This behavior
may change in the future.
  plt.scatter(X_test[:, 0], X_test[:, 1], c=y_test, cmap='coolwarm',
edgecolors='k', marker='x', label='Test')
```

```
<matplotlib.collections.PathCollection at 0x7b0365c47590>
```



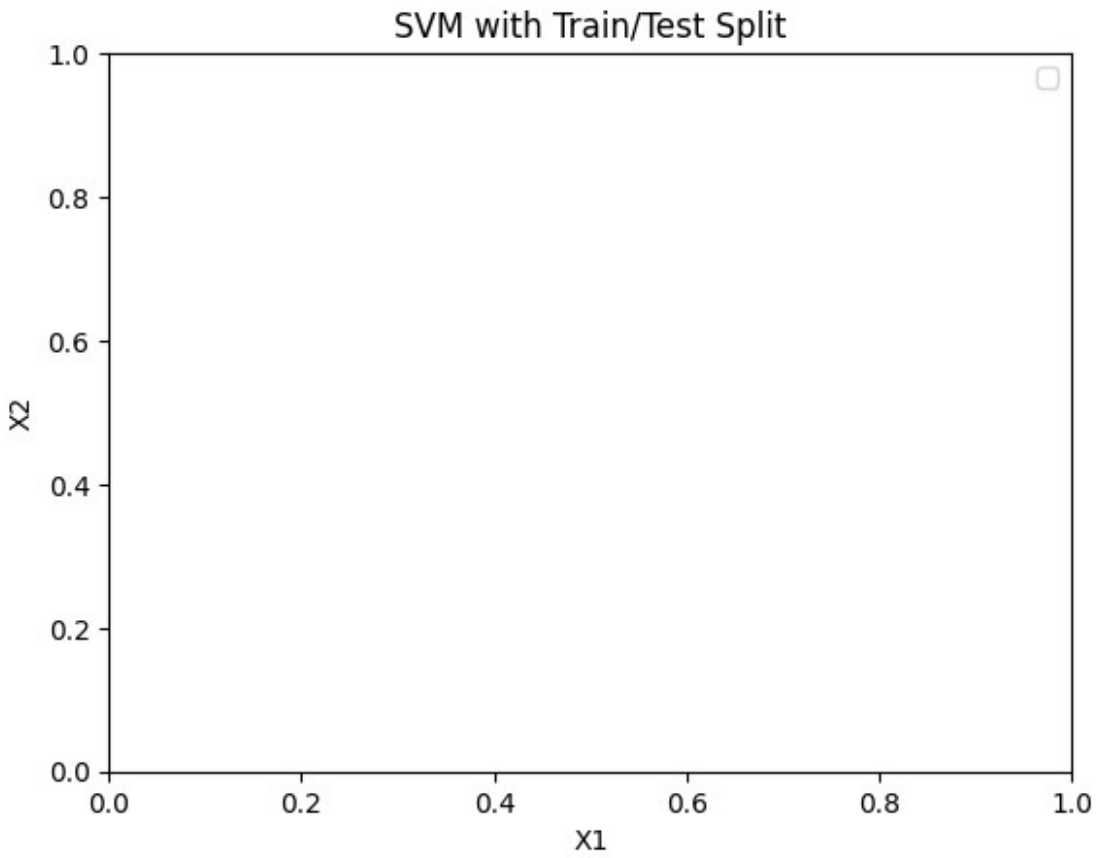

```
# Plot decision boundary
ax = plt.gca()
xlim = ax.get_xlim()
ylim = ax.get_ylim()
```




```
xx = np.linspace(xlim[0], xlim[1])
yy = np.linspace(ylim[0], ylim[1])
YY, XX = np.meshgrid(yy, xx)
xy = np.vstack([XX.ravel(), YY.ravel()]).T
Z = model.decision_function(xy).reshape(XX.shape)

ax.contour(XX, YY, Z, colors='black', levels=[-1, 0, 1],
linestyles=['--', '-', '--'])
plt.title('SVM with Train/Test Split')
plt.xlabel('X1')
plt.ylabel('X2')
plt.legend()
plt.show()
```

```
<ipython-input-13-ebe931b8d038>:5: UserWarning: No artists with labels
found to put in legend.  Note that artists whose label start with an
underscore are ignored when legend() is called with no argument.
  plt.legend()
```


SVM with Train/Test Split
1.0
0.8
0.6
0.4
0.2
0.0
X2
0.0
0.2
0.4
0.6
0.8
1.0
X1