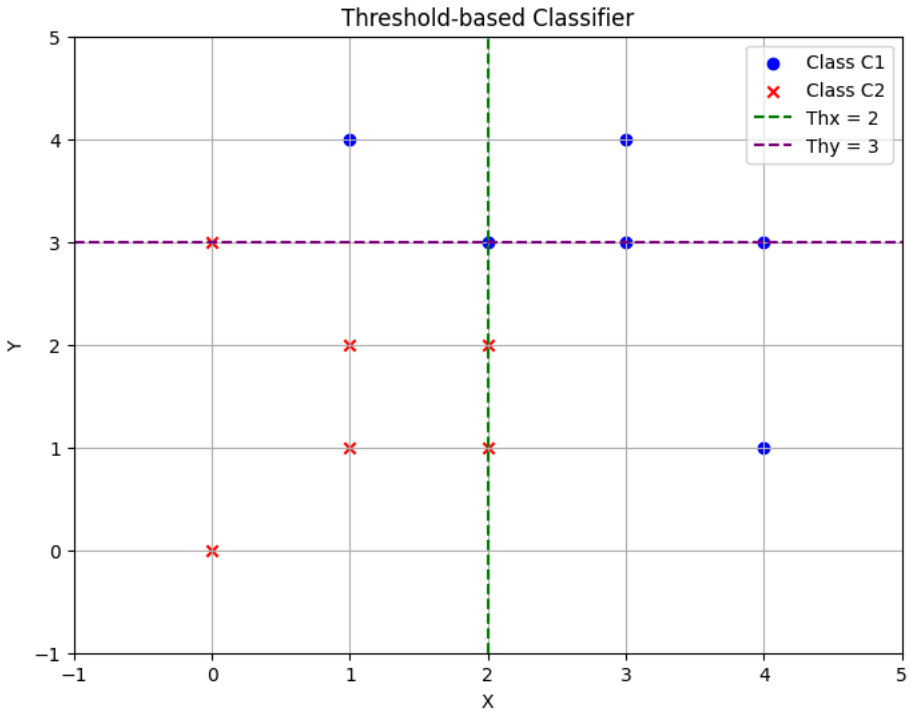
```
import matplotlib.pyplot as plt

# Define the data samples for classes C1 and C2
C1 = [(2, 3), (3, 3), (3, 4), (1, 4), (4, 1), (4, 3)]
C2 = [(0, 0), (0, 3), (1, 1), (1, 2), (2, 1), (2, 2)]


def plot_data(thx, thy):
    """Plot the data samples with threshold lines."""
    plt.figure(figsize=(8, 6))
    plt.scatter(*zip(*C1), color='blue', marker='o', label='Class C1')
    plt.scatter(*zip(*C2), color='red', marker='x', label='Class C2')
    plt.axvline(x=thx, color='green', linestyle='--', label=f'Thx = {thx}')
    plt.axhline(y=thy, color='purple', linestyle='--', label=f'Thy = {thy}')
    plt.xlim([-1, 5])
    plt.ylim([-1, 5])
    plt.xlabel('X')
    plt.ylabel('Y')
    plt.title('Threshold-based Classifier')
    plt.legend()
    plt.grid(True)
    plt.show()


def calculate_accuracy(thx, thy):
    """Classify data points and calculate the classification accuracy."""
    correct = sum(1 for x, y in C1 if x >= thx and y >= thy)
    correct += sum(1 for x, y in C2 if x < thx or y < thy)
    return (correct / 12) * 100


# From testing the best thresholds turned out to be 2 and 3
best_thx = 2
best_thy = 3
best_accuracy = calculate_accuracy(best_thx, best_thy)

print(f"Best thresholds: Thx = {best_thx}, Thy = {best_thy}")
print(f"Highest classification accuracy: {best_accuracy:.2f}%")
plot_data(best_thx, best_thy)
```

```
Best thresholds: Thx = 2, Thy = 3
Highest classification accuracy: 83.33%
```




Colab Link: https://colab.research.google.com/drive/1bORy-2Y2vQfIDwxaZLtc4I3X_pHsRpXX?usp=sharing