



```
import matplotlib.pyplot as plt

def plot_data_samples(class1, class2, thx, thy):
    plt.scatter(*zip(*class1), label='Class C1', marker='o')
    plt.scatter(*zip(*class2), label='Class C2', marker='s')
    plt.axvline(x=thx, color='r', linestyle='--', label=f'thx = {thx}')
    plt.axhline(y=thy, color='b', linestyle='--', label=f'thy = {thy}')
    plt.xlabel('X-axis')
    plt.ylabel('Y-axis')
    plt.legend()
    plt.xlim([-1, 5])
    plt.ylim([-1, 5])
    plt.title('Data Samples and Thresholds')
    plt.show()

def get_threshold(prompt):
    while True:
        try:
            value = float(input(prompt))
            return value
        except ValueError:
            print("Invalid input. Please enter a number.")

def classify_data_samples(data, thx, thy):
    class1_count = sum(1 for x, y in data if x >= thx and y >= thy)
    class2_count = len(data) - class1_count
    return class1_count, class2_count

def calculate_accuracy(class1_count, class2_count):
    total_samples = class1_count + class2_count
    accuracy = class1_count / total_samples
    return accuracy

#data samples
class1_data = [(2, 3), (3, 3), (3, 4), (1, 4), (4, 1), (4, 3)]
class2_data = [(0, 0), (0, 3), (1, 1), (1, 2), (2, 1), (2, 2)]
```

```

while True:
    #plot data samples
    plot_data_samples(class1_data, class2_data, 0, 0)

    #get user enter threshold
    thx = get_threshold("Enter the threshold for x (or 'x' to exit): ")
    if thx == 'x':
        break

    thy = get_threshold("Enter the threshold for y (or 'x' to exit): ")
    if thy == 'x':
        break

    #classify data samples and calculate accuracy
    class1_count, class2_count = classify_data_samples(class1_data + class2_data, thx, thy)
    accuracy = calculate_accuracy(class1_count, class2_count)

    #training accuracy
    print(f"Training Accuracy: {accuracy * 100:.2f}%")

    #data samples plot
    plot_data_samples(class1_data, class2_data, thx, thy)

print("Suggested Thresholds for Highest Accuracy:")
print("thx = 2.5, thy = 1.5 (for example)")

```

