Jenna Leali Assignment 3 CAP 4613

https://colab.research.google.com/drive/11OIgeVE9WyOWoQg2fhUDJT79Ld-Xvxl9?usp=sharing

Threshold-based Classifier. We have a two-class classification problem (i.e., C1 and C2). Each data sample is represented by two attributes (x, y). The three data samples in class C1 are {(2, 3), (3, 3), (3, 4), (1, 4), (4, 1), (4, 3)} and {(0, 0), (0, 3), (1, 1), (1, 2), (2, 1), (2, 2)} in class C2. Perform the following in Python:

8) Based on your observation from the results above, what is a suitable set of thresholds that will give the highest accuracy? Report your suggested thresholds and the corresponding classification accuracy.

Based on my observations, the best threshold values are thx=1.0 and thy=3.0, which have a high classification accuracy of 91.67%. This set of thresholds means that the classifier correctly identifies 11 out of the 12 data points in the dataset.

1) Plot the data samples. The data points in classes C1 and C2 must be in two different colors and shapes. Label the axes and add legends as appropriate.

```
import matplotlib.pyplot as plt
```

```
def plotC1C2(C1, C2):
    c1_x = [point[0] for point in C1]
    c1_y = [point[1] for point in C1]
    c2_x = [point[0] for point in C2]
    c2_y = [point[1] for point in C2]

    plt.scatter(c1_x, c1_y, color='blue', marker='o', label='Class C1')
    plt.scatter(c2_x, c2_y, color='red', marker='x', label='Class C2')

    plt.xlabel('X-axis')
    plt.ylabel('Y-axis')
    plt.legend()
    plt.xlim(-1, 5)
    plt.ylim(-1, 5)
    plt.grid(True)
    plt.show()
```

2) The code asks the user to enter two thresholds thx and thy.

3) The code must use appropriate conditions to prevent common errors such as entering characters instead of numbers etc.

7) The program will exit the loop once the user writes 'x' as an input for either threshold.

```
def thresholds():
    while True:
        try:
            thx = input("Enter threshold thx or enter x to exit: ")
            if thx.lower() == 'x':
                return 'x', 'x'
            thx = float(thx)

            thy = input("Enter threshold thy or enter x to exit: ")
            if thy.lower() == 'x':
                return 'x', 'x'
            thy = float(thy)

            return thx, thy
        except ValueError:
            print("Please enter valid numbers.")
```

4) Your code calculates and prints the training accuracy based on the user-entered thresholds. To do so, assume that for any data point (x, y) with x>= thx and y>= thy, the data sample belongs to class C1, and C2 if otherwise. Using this rule and the user-entered thresholds, the code calculates the classification accuracy for the six data samples. The classification accuracy is defined as the number of correctly classified data points divided by the total number of data points (12 here).

```
def calculate_accuracy(C1, C2, thx, thy):
    correct = 0
    for x, y in C1:
        if x >= thx and y >= thy:
            correct += 1
    for x, y in C2:
        if x < thx or y < thy:
            correct += 1
    accuracy = correct / 12
    return accuracy
```

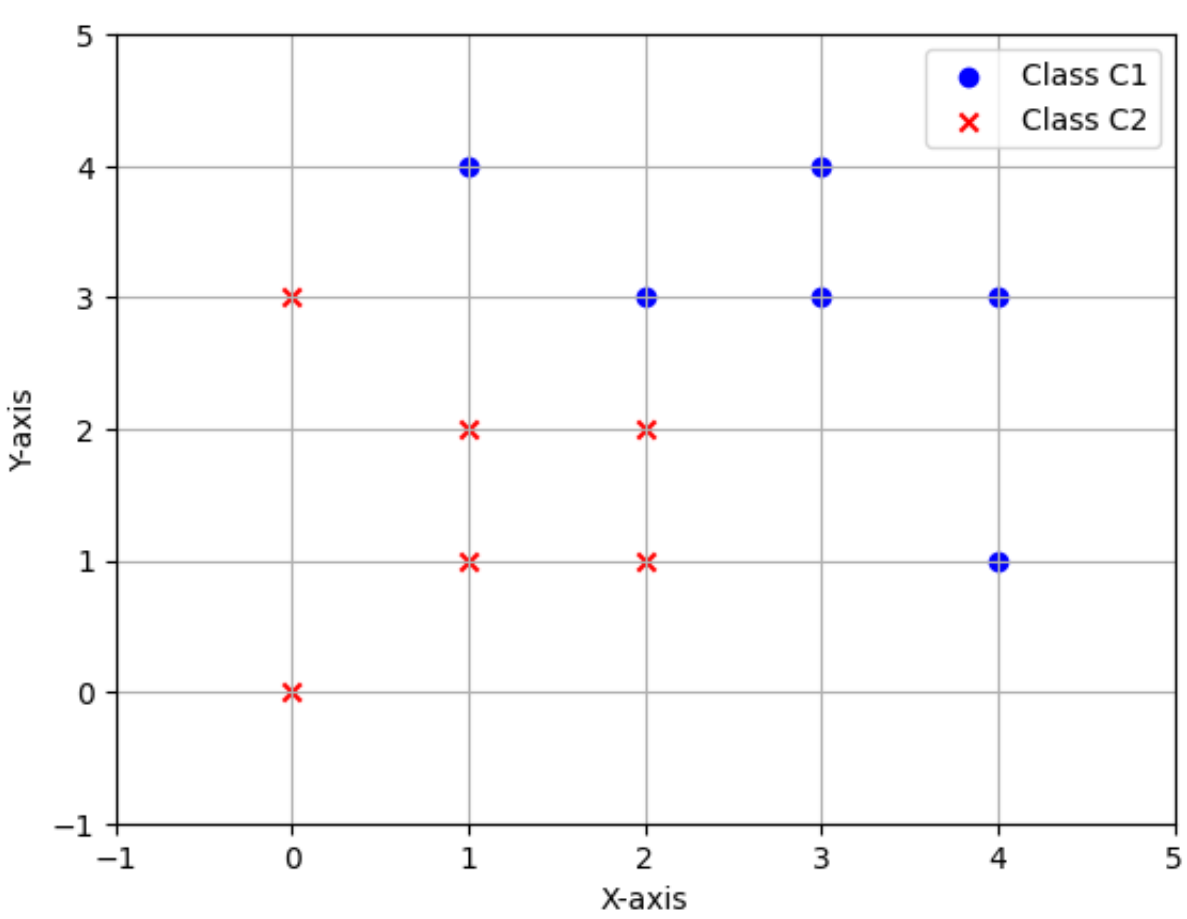
5) Plot the data samples, the selected thx, and the thx lines.

```
def data_thresholds(C1, C2, thx, thy):
    for x, y in C1:
        plt.scatter(x, y, color='blue', marker='o', label='Class C1')

    for x, y in C2:
        plt.scatter(x, y, color='red', marker='x', label='Class C2')

    plt.axvline(x=thx, color='orange', linestyle='--', label=f'Threshold thx={thx
    plt.axhline(y=thy, color='orange', linestyle='--', label=f'Threshold thy={thy

    plt.xlabel('X-axis')
    plt.ylabel('Y-axis')

    plt.xlim(-1, 5)
    plt.ylim(-1, 5)

    plt.legend([plt.Line2D([0], [0], color='blue', marker='o', linestyle=''),
                plt.Line2D([0], [0], color='red', marker='x', linestyle=''),
                plt.Line2D([0], [0], color='orange', linestyle='--')],
               ['Class C1', 'Class C2', f'Threshold thx={thx}', f'Threshold thy={

    plt.grid(True)
    plt.show()
```

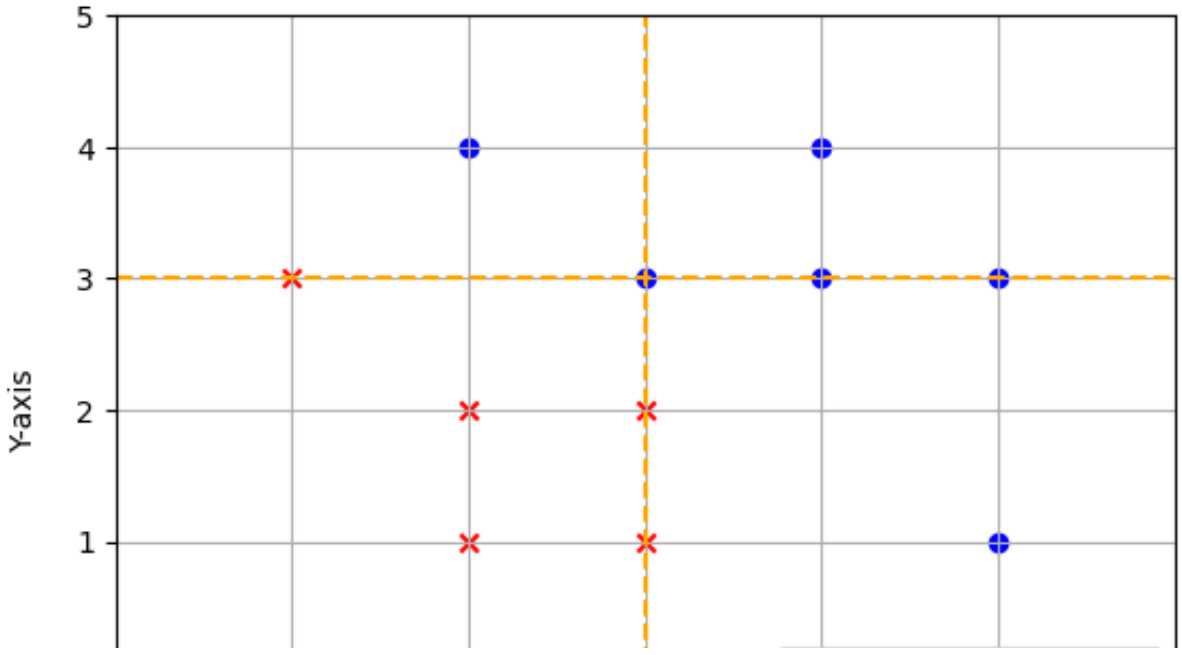
6) Create a loop to continuously repeat parts (d) to (e) and enter different sets of thresholds each time.

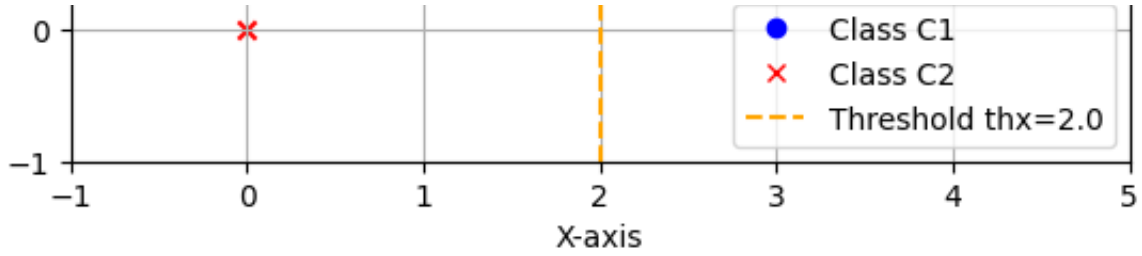
```
def main():
    C1 = [(2, 3), (3, 3), (3, 4), (1, 4), (4, 1), (4, 3)]
    C2 = [(0, 0), (0, 3), (1, 1), (1, 2), (2, 1), (2, 2)]

    plotC1C2(C1, C2)

    while True:
        thx, thy = thresholds()
        if thx == 'x' or thy == 'x':
            print("Exiting program.")
            break

        thx, thy = float(thx), float(thy)
        accuracy = calculate_accuracy(C1, C2, thx, thy)
        print(f"Classification accuracy with thresholds ({thx}, {thy}): {accuracy
```

```
        data_thresholds(C1, C2, thx, thy)
main()
```

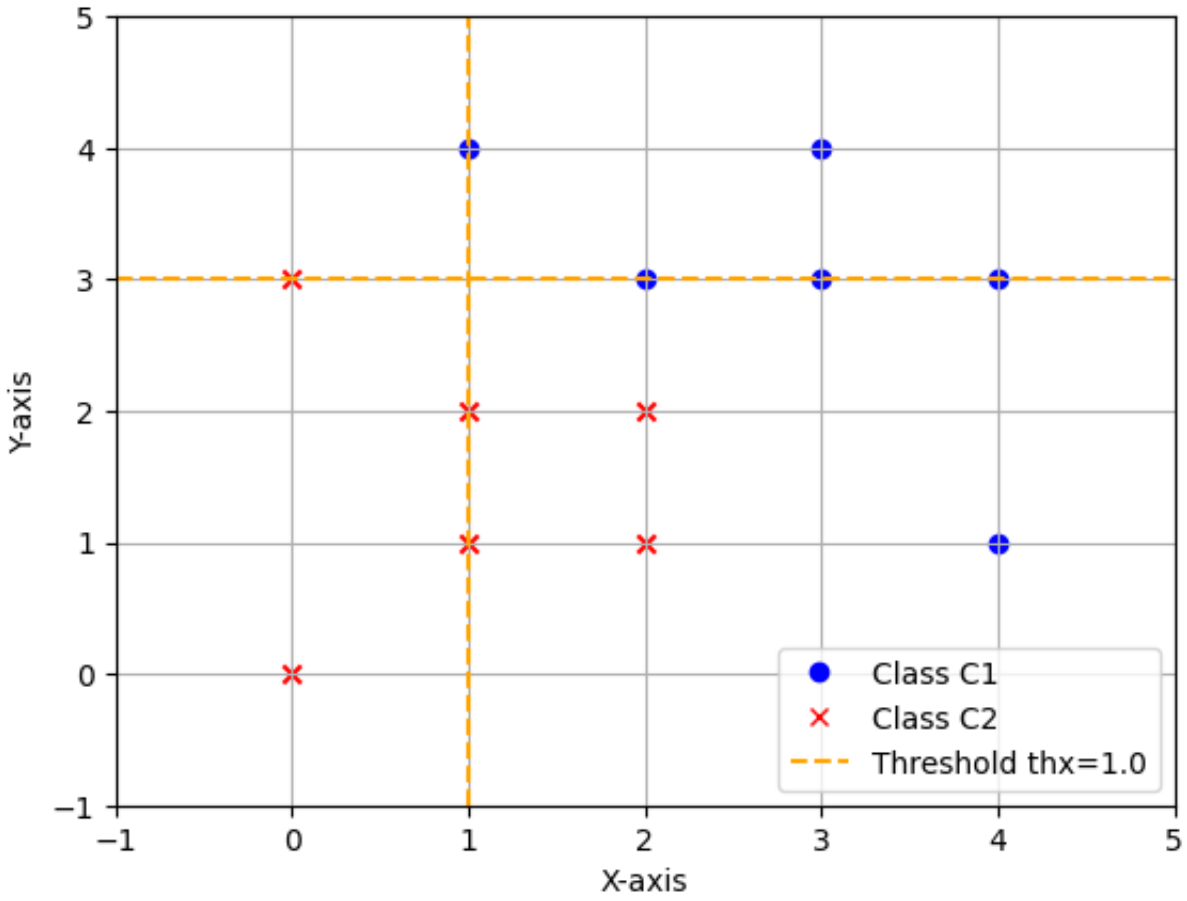



```
Enter threshold thx or enter x to exit: 2
Enter threshold thy or enter x to exit: 3
Classification accuracy with thresholds (2.0, 3.0): 83.33%
```

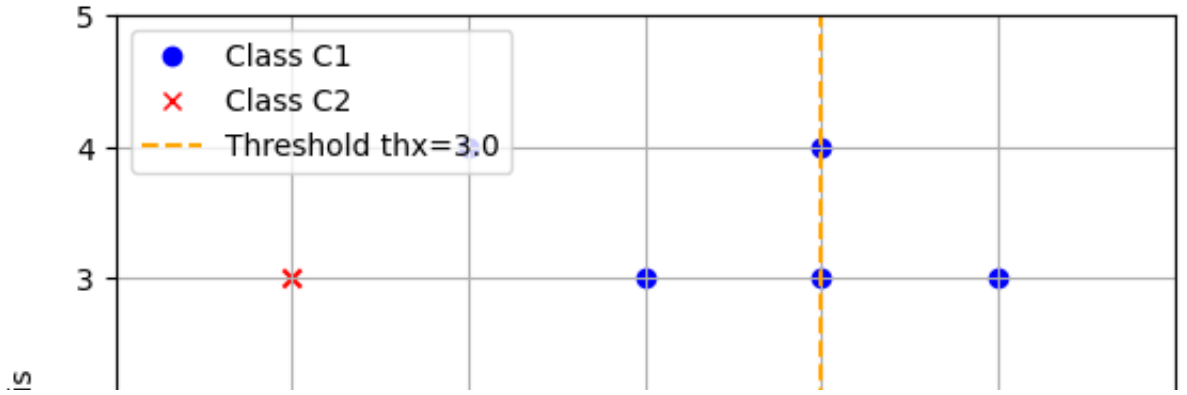


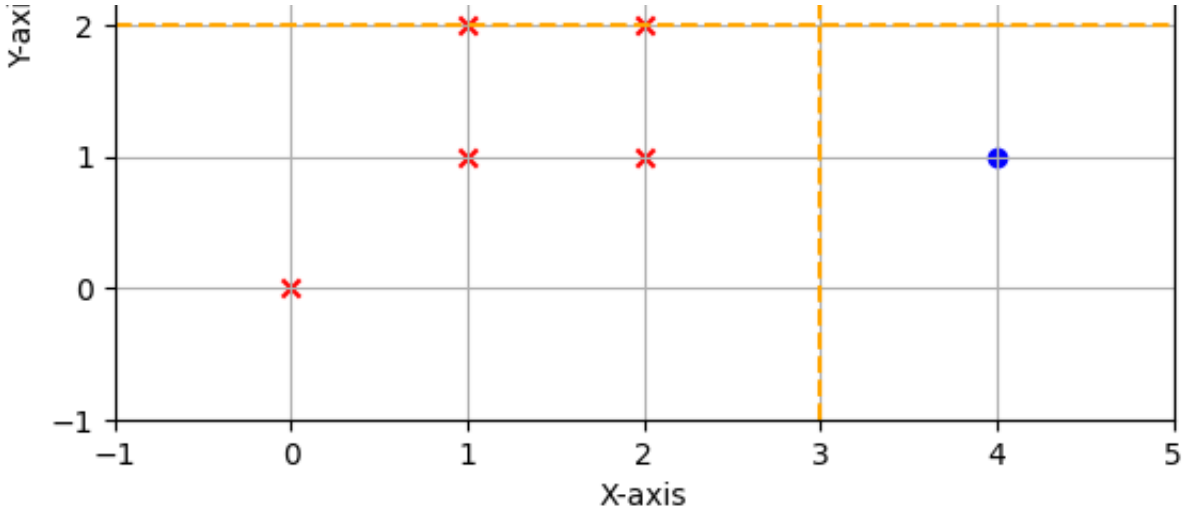



```
Enter threshold thx or enter x to exit: 1
Enter threshold thy or enter x to exit: 3
Classification accuracy with thresholds (1.0, 3.0): 91.67%
```




```
Enter threshold thx or enter x to exit: 3
Enter threshold thy or enter x to exit: 2
Classification accuracy with thresholds (3.0, 2.0): 75.00%
```






```
Enter threshold thx or enter x to exit: x
Exiting program.
```

https://colab.research.google.com/drive/11OIgeVE9WyOWoQg2fhUDJT79Ld-Xvxl9?usp=sharing