Assignment 1

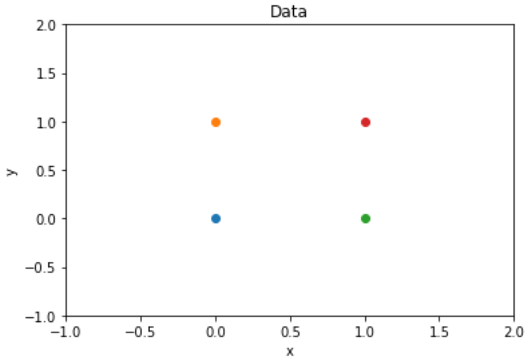
Name: Morgan Benavidez

Z-Number: 23589091

Link: https://colab.research.google.com/drive/1y4e1fSnHOo8OVs5QUl5NHdleHHNTnfQN?usp=sharing

Problem 1

```
import matplotlib.pyplot as plt


inputSamples = [(0,0), (0,1), (1,0), (1,1)]
for i in range(0, len(inputSamples)):

  plt.scatter(inputSamples[i][0], inputSamples[i][1])

plt.ylabel('y')
plt.xlabel('x')
plt.title('Data')
plt.axis([-1, 2, -1, 2])
plt.show()
```



```
def and_operation(x, y):

  if x == 0 and y == 0:
    output = 0
  elif x == 0 and y == 1:
    output = 0
  elif x == 1 and y == 0:
    output = 0
  elif x == 1 and y == 1:
    output = 1
  else:
    output = 'x and y must be a 0 or 1'
  return output

```

```
def or_operation(x, y):

  if x == 0 and y == 0:
    output = 0
  elif x == 0 and y == 1:
    output = 1
  elif x == 1 and y == 0:
    output = 1
  elif x == 1 and y == 1:
    output = 1
  else:
    output = 'x and y must be a 0 or 1'
  return output

```

```
def xor_operation(x, y):

  if x == 0 and y == 0:
    output = 0
  elif x == 0 and y == 1:
    output = 1
  elif x == 1 and y == 0:
    output = 1
  elif x == 1 and y == 1:
    output = 0
  else:
    output = 'x and y must be a 0 or 1'
  return output

```

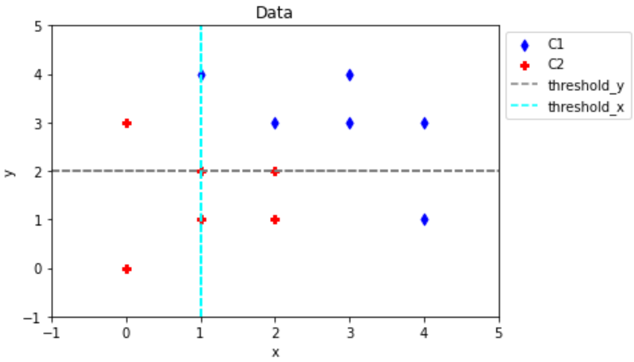
```
# Test for functions using input Samples

and_list = []
or_list = []
xor_list = []

for i in range(0, len(inputSamples)):
  x = inputSamples[i][0]
  y = inputSamples[i][1]

  #print('x = ' + str(x) + ', y = ' + str(y))

  and_list.append(and_operation(x,y))

  or_list.append(or_operation(x,y))

  xor_list.append(xor_operation(x,y))

print('AND List Output: ' + str(and_list))
print('OR List Output: ' + str(or_list))
print('XOR List Output: ' + str(xor_list))
```

```
AND List Output: [0, 0, 0, 1]
OR List Output: [0, 1, 1, 1]
XOR List Output: [0, 1, 1, 0]
```

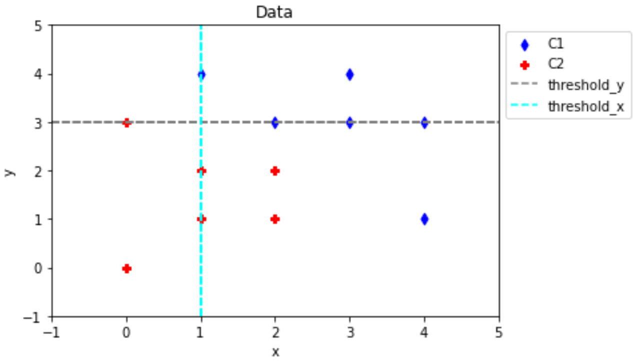
Problem 2

```
import numpy as np
import matplotlib.pyplot as plt
import time



def classification_accuracy(threshold_x, threshold_y, C1, C2):

  correct = 0
  incorrect = 0
  total = len(C1) + len(C2)
  print(total)

  for i in range(0, len(C1)):

    x = C1[i][0]
    y = C1[i][1]
    x2 = C2[i][0]
    y2 = C2[i][1]

    if (x >= threshold_x and y >= threshold_y):
      correct += 1
    else:
      incorrect += 1
    if (x2 >= threshold_x and y2 >= threshold_y):
      incorrect += 1
    else:
      correct += 1
  return correct, incorrect, total

def obtain_thresholds():

  while (True):
    try:
      var = input("Threshold x must be a number, please enter an integer: ")
      if (var == 'x'):
        testing = False
        threshold_x = 'x'
        break
      threshold_x = int(var)
    except ValueError:
      print("Threshold must be an integer, please try again.")
      continue
    else:
      break

  while (True):
    try:
      var2 = input("Threshold y must be a number, please enter an integer: ")
      if (var2 == 'x'):
        testing = False
        threshold_y = 'x'
        break
      threshold_y = int(var2)
    except ValueError:
      print("Threshold must be an integer, please try again.")
      continue
    else:
      break

  return threshold_x, threshold_y

def print_accuracy_results(results):

  print('Correct: ' + str(results[0]))
  print('Incorrect: ' + str(results[1]))
  print('Total: ' + str(results[2]))
  print('Classification Accuracy: ' + str(results[0]/results[2]))


def create_plot(C1, C2, threshold_x, threshold_y):

  for i in range(0, len(C1)):

    plt.scatter(C1[i][0], C1[i][1], color='blue', marker='d')
    plt.scatter(C2[i][0], C2[i][1], color='red', marker='P')
    plt.hlines(y=threshold_y, xmin=-1, xmax=5, linestyle='dashed', color='gray')
    plt.vlines(x=threshold_x, ymin=-1, ymax=5, linestyle='dashed', color='cyan')

  plt.ylabel('y')
  plt.xlabel('x')
  plt.title('Data')
  plt.axis([-1, 5, -1, 5])
  plt.legend(['C1', 'C2', 'threshold_y', 'threshold_x'], bbox_to_anchor=(1, 1), loc='upper left')
  plt.show()
  print('\n')

def main(C1, C2):

  thresholds = obtain_thresholds()
  threshold_x = thresholds[0]
  threshold_y = thresholds[1]

  if (threshold_x == 'x' or threshold_y == 'x'):
    return 'x'
  else:
    results = classification_accuracy(threshold_x, threshold_y, C1, C2)
    print_accuracy_results(results)
    create_plot(C1, C2, threshold_x, threshold_y)
    return 'Please enter another set of thresholds, or x to quit.'


C1 = np.array([(2,3), (3,3), (3,4), (1,4), (4,1), (4,3)])
C2 = np.array([(0,0), (0,3), (1,1), (1,2), (2,1), (2,2)])

testing = True
while (testing == True):
  message = main(C1, C2)
  if (message == 'x'):
    testing = False
    print('User Terminated Program')
  else:
    print(message)

```

```
Threshold x must be a number, please enter an integer: 1
Threshold y must be a number, please enter an integer: 2
12
Correct: 9
Incorrect: 3
Total: 12
Classification Accuracy: 0.75
```



```
Please enter another set of thresholds, or x to quit.
Threshold x must be a number, please enter an integer: 1
Threshold y must be a number, please enter an integer: 3
12
Correct: 11
Incorrect: 1
Total: 12
Classification Accuracy: 0.9166666666666666
```



```
Please enter another set of thresholds, or x to quit.
Threshold x must be a number, please enter an integer: x
Threshold y must be a number, please enter an integer: 1
User Terminated Program
```

The set of thresholds that will yield the highest classification accuracy is threshold_x = 1 and threshold_y = 3. The classification accuracy for this set of thresholds is 91.67%.

✓ 0s completed at 5:12 AM ● ×