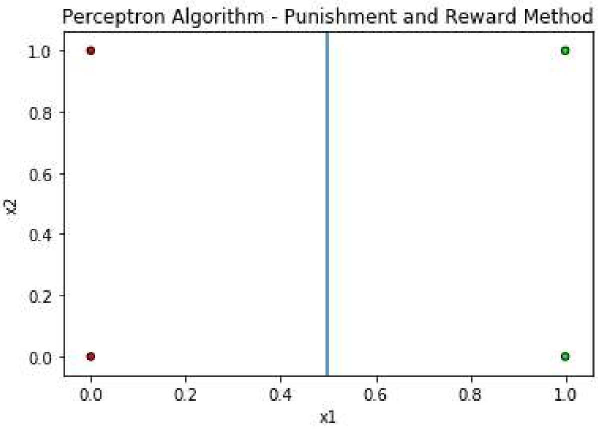
Homework 2 : Problem 3.4

Consider a case in which class 1 consists of the two feature vectors [0, 0] and [0, 1] and class 2 of [1, 0] and [1, 1]. Use the perceptron algorithm in its reward and punishment form, with rho = 1 and w(0) = [0, 0], to design the line separating the two classes.

```
# -*- coding: utf-8 -*-
"""
Homework 2 - Problem 3.4

Perceptron Algorithm
@author: Nazneen Kotwal
"""

import os
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap


def decbound(w):
    """
    To plot the decision boundary
    Parameters
    ----------
    arg1 : wht
         Weights associated witht the feature vector
         Format: [w0,w1,w2]
    """
    m = -(w[0]/w[1])
    plt.axvline(m)
    plt.title("Perceptron Algorithm - Punishment and Reward Method")
    plt.xlabel("x1")
    plt.ylabel("x2")

def percepalgo(x,w,rho):
    """
    Perceptron Algorithm for Punishment and Reward Method
    ----------
    arg1 : trainingSet
        Feature vector
    arg2 : wht
        Weights associated witht the feature vector
        Format [w0,w1,w2]
    arg3 : rho
        Controls the rate of convergence of the algorithm
    """
    if ((np.dot(w,x[0:3]) <= 0) and (x[-1] == 0)):
        mul =  rho * x[0:3]
        w = np.add(w,mul)
    elif ((np.dot(w,x[0:3]) >= 0) and (x[-1] == 1)):
        mul =  rho * x[0:3]
        w = np.subtract(w,mul)
    else:
        w = w
    return(w)

def main():
    path = "C:/Users/nazne/OneDrive/Documents/1_ECE 759 Pattern Recognition/Homework/Homework2/proble
    print ("The current working directory is", os.getcwd())
    os.chdir(path)
    x1 = []
    x2 = []
    x3 = []
    x = [[1,0,0,0],[1,0, 1, 0],[1,1,0,1],[1,1,1,1]]
    winit = [0 ,0, 0]
    for i in range(len(x)):
        x1.append(x[i][1])
        x2.append(x[i][2])
        x3.append(x[i][3])
    cmap_bold = ListedColormap(['#FF0000', '#00FF00'])
    plt.scatter(x1, x2, c=x3, cmap=cmap_bold,
                edgecolor='k', s=20)
    rho = 1
    count = 0;
    while True:
        flag = 0
        count += 1
        for i in range(len(x)):
            wht = percepalgo(x[i],winit,rho)
            if (np.array_equal(winit,wht)==False):
                flag = 1
                winit = wht
        if (flag == 0) or (count == 100):
            break
    decbound(wht)
    print('The Update Weight Vector on convergence w = [w0 w1 w2] is %s: ' %(wht))
    print('Number of Loops untill Covergence: %d'  % count)

if __name__ == '__main__':
    main()
```

Note:

- The updated weight vector on convergence: w = [w0 w1 w2] is [1 -2 0]
- Number of loops untill covergence: 4


Perceptron Algorithm - Punishment and Reward Method
x2
x1