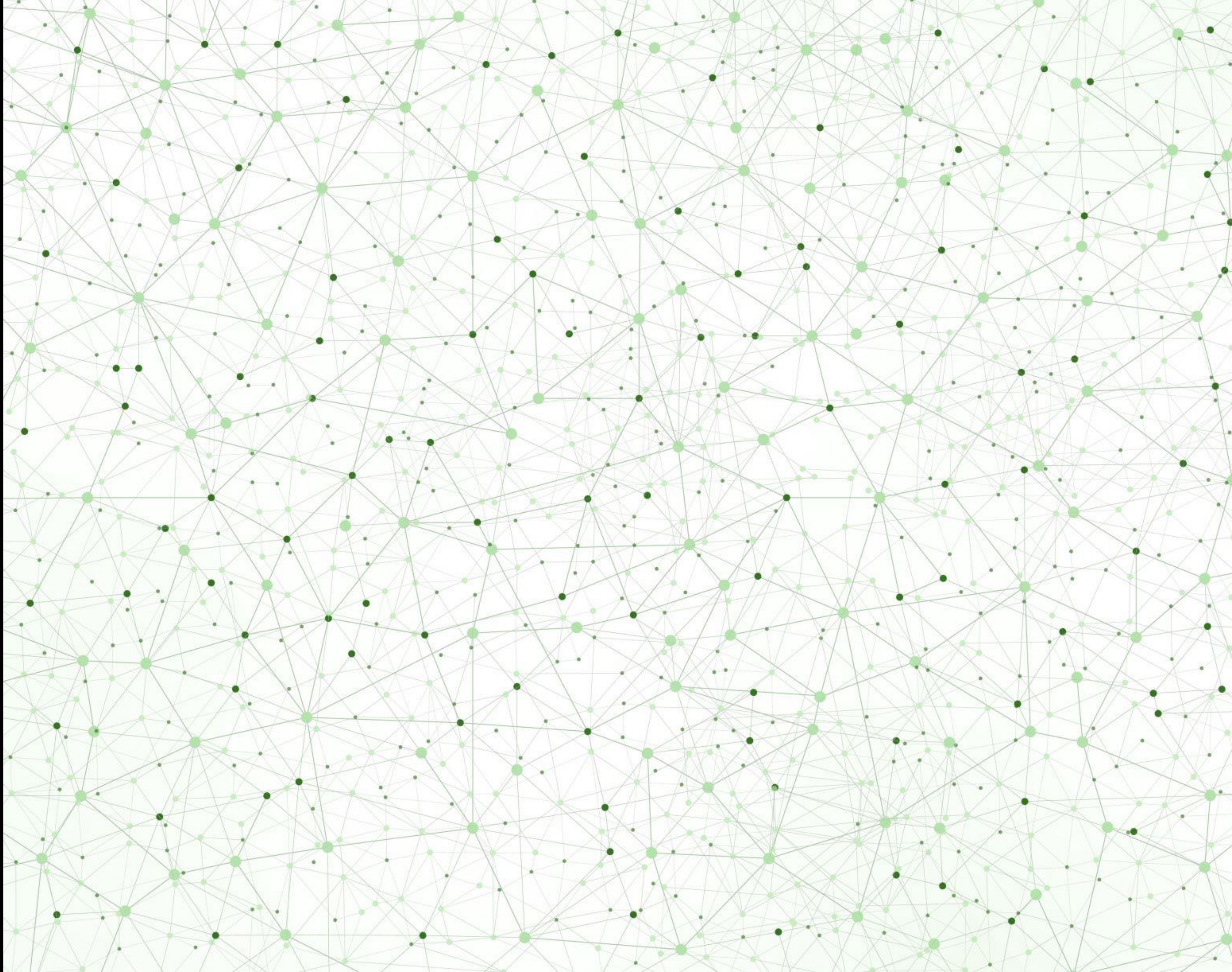
ML2: Perceptron

Manuel, Ma. Janelle G.

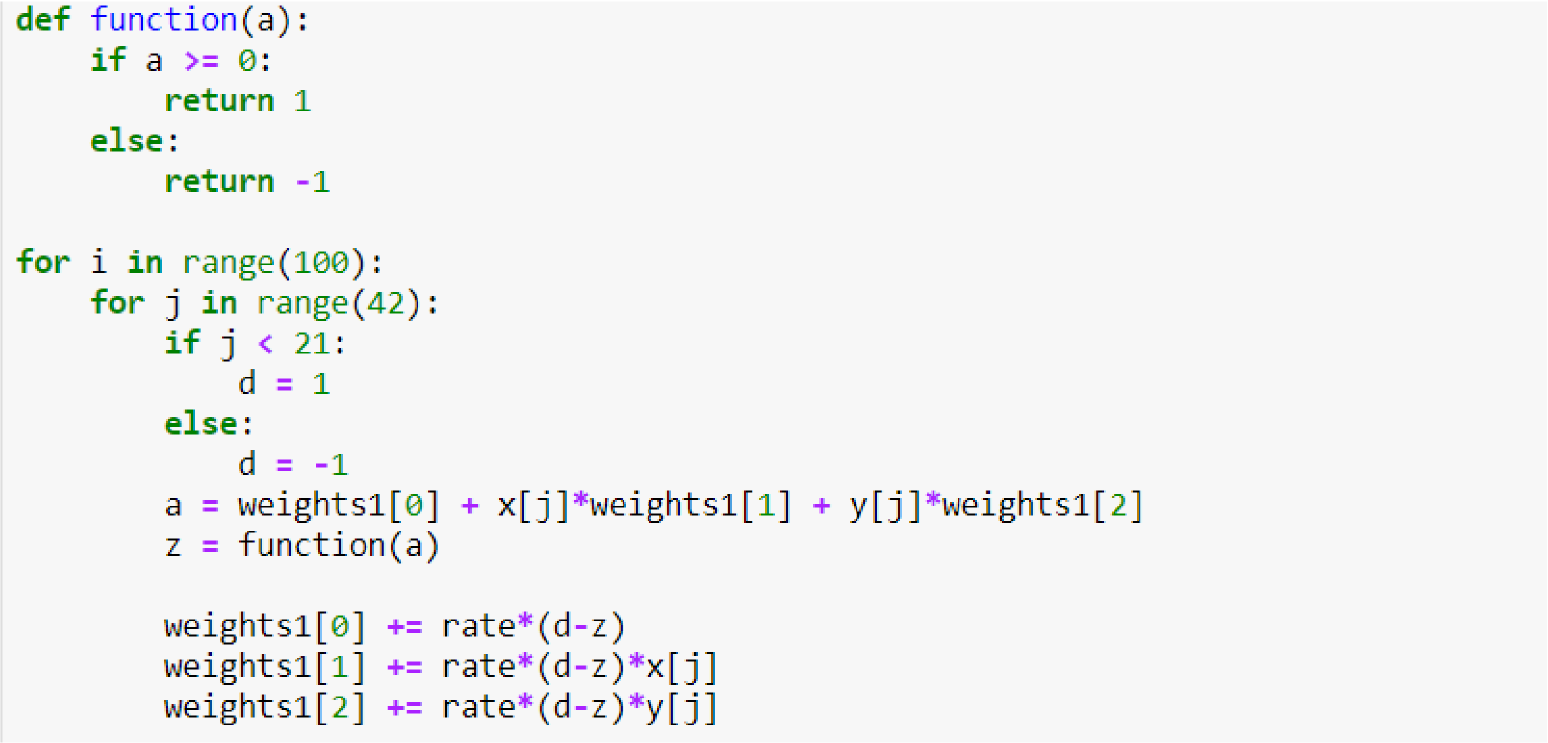
AP 186



Objectives:

The main objective of this activity is to compute the decision surface between the two classes using the **perceptron algorithm**.

Results: Perceptron Algorithm

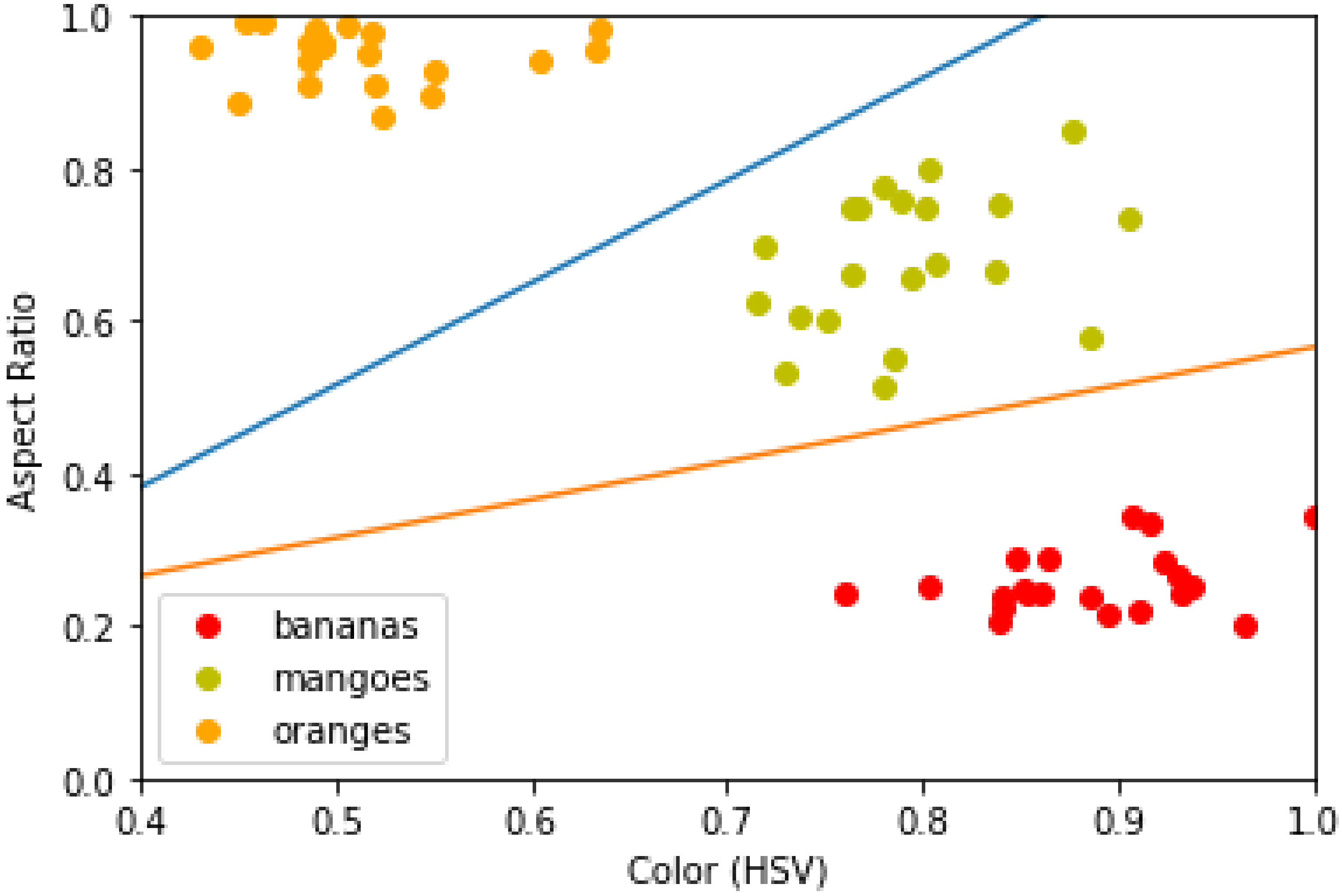


```python
def function(a):
    if a >= 0:
        return 1
    else:
        return -1

for i in range(100):
    for j in range(42):
        if j < 21:
            d = 1
        else:
            d = -1
        a = weights1[0] + x[j]*weights1[1] + y[j]*weights1[2]
        z = function(a)

        weights1[0] += rate*(d-z)
        weights1[1] += rate*(d-z)*x[j]
        weights1[2] += rate*(d-z)*y[j]
```

Figure 2: Simple Perceptron Algorithm in Python



Blue decision line (mango vs orange):
y= 1.3451233019364752 x + 0.1571489100747912

Orange decision line (mango vs bananas):
y= 0.5001057568800451 x + 0.06555828479948116

Figure 2: Decision lines for the three fruit classes

Discussion

The decision lines for each pair of classes were successfully obtained and plotted as shown in Figure 2. Blue decision line separates the mangoes from the oranges while the orange decision line separates the mangoes from the bananas.

The **initial weights** that I used for both decisions lines are **w0= 0.1, w1=0.2, w2=-0.2**.

The **final weights** for orange vs mango decision line are

w0= 0.3854543025001047, w1= -3.299313777472603, and w2=2.4527965374793697.

The **final weights** for mango vs banana decision line are

w0= -0.26170127453084074, w1= -1.9963657434911168, and w2= 3.991887147921721.

Testing the perceptron

Mango vs Orange

W0= -0.3854543025001047

W1= -3.299313777472603

W2=2.4527965374793697

If x1= 0.48 and x2=0.9 (features of an orange), the feature point is equal to

0.23873 (above the blue decision line!)

Mango vs Banana

W0= -0.26170127453084074

W1= -1.9963657434911168

W2= 3.991887147921721

If x1= 0.84 and x2=0.2 (features of a banana), the feature point is equal to

-1.14 (below the orange decision line!)

Analysis

Upon testing the perceptron more, the algorithm accurately classified the fruits from each other.

To produce a decision line/separation that is somewhat more equidistant from the different clusters, we can increase the iterations so that (d-z) is really small.

Reflection

Rating 10/10

I was able to finish this activity successfully! I am quite sure with my algorithm since I've checked it for numerous values of x1 and x2. This activity was quite intimidating but it was actually pretty easy.

References

- Soriano, M. (2020). ML2 – Perceptron.
- Python Machine Learning Tutorial. Machine Learning with Python: Separating Classes with Dividing Lines (python-course.eu)