3.

a. Support Vectors:
[-0.5 -0.5]
[0.2 0.8]
[2.0 -1.0]

b. Weight Vector: [1.0 1.0]
Bias: 0.0

w = c - d = [0.49982183 0.50013169] - [-0.49997676 -0.49997676]

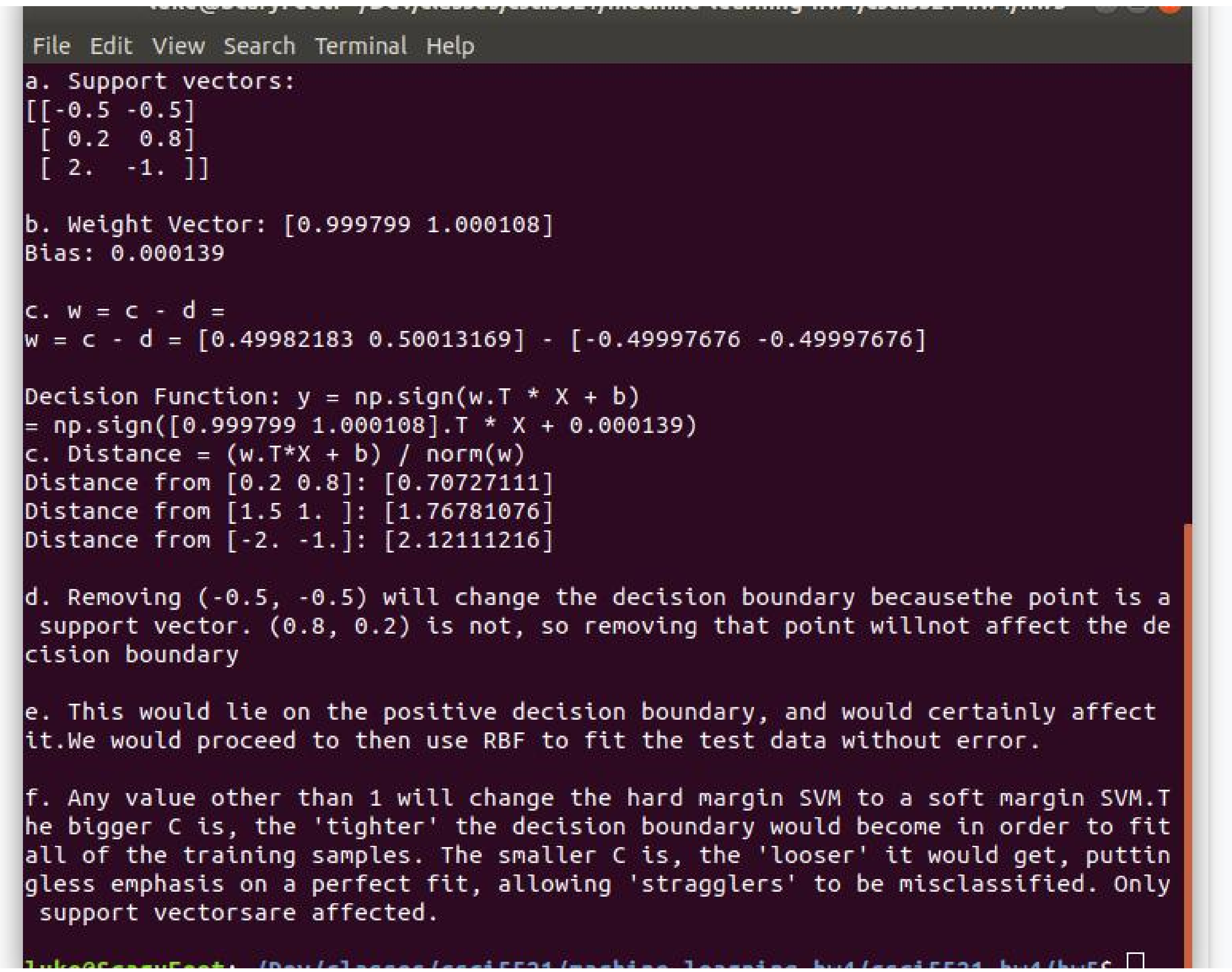
Decision Function: y = np.sign(w.T * X + b)
= np.sign([0.999799 1.000108].T * X + 0.000139)

c. Distance = (w.T*X + b) / norm(w)
Distance from [0.2 0.8]: [0.70727111]
Distance from [1.5 1.]: [1.76781076]
Distance from [-2. -1.]: [2.12111216]

d. Removing (-0.5, -0.5) will change the decision boundary because the point is a support vector. (0.8, 0.2) is not, so removing that point will not affect the decision boundary

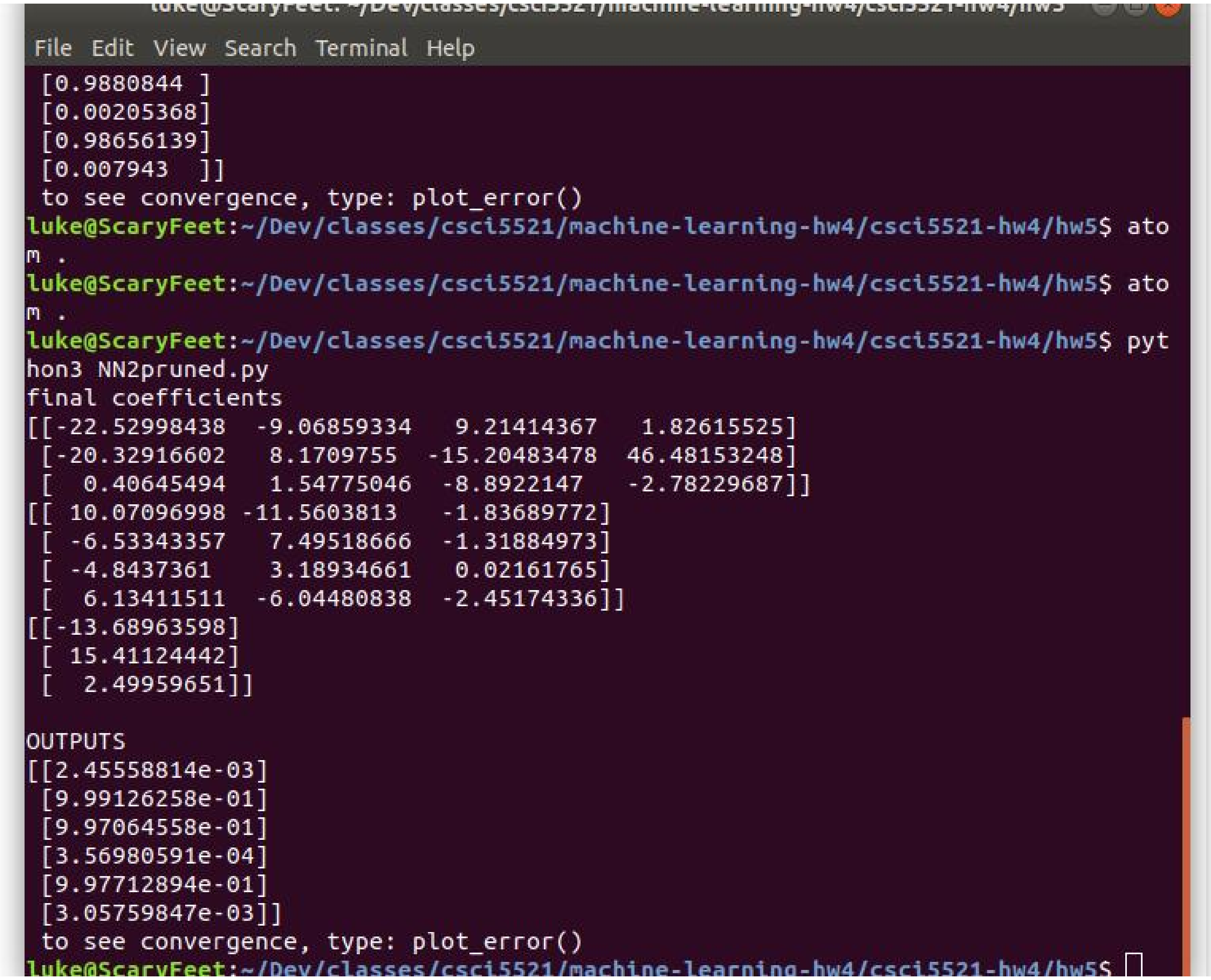
e. This would lie on the positive decision boundary, and would certainly affect it.We would proceed to then use RBF to fit the test data without error.

f. Any value other than 1 will change the hard margin SVM to a soft margin SVM.The bigger C is, the 'tighter' the decision boundary would become in order to fit all of the training samples. The smaller C is, the 'looser' it would get, putting less emphasis on a perfect fit, allowing 'stragglers' to be misclassified. Only support vectors are affected.



```
a. Support vectors:
[[-0.5 -0.5]
 [ 0.2  0.8]
 [ 2.  -1. ]]

b. Weight Vector: [0.999799 1.000108]
Bias: 0.000139

c. w = c - d =
w = c - d = [0.49982183 0.50013169] - [-0.49997676 -0.49997676]

Decision Function: y = np.sign(w.T * X + b)
= np.sign([0.999799 1.000108].T * X + 0.000139)
c. Distance = (w.T*X + b) / norm(w)
Distance from [0.2 0.8]: [0.70727111]
Distance from [1.5 1. ]: [1.76781076]
Distance from [-2. -1.]: [2.12111216]

d. Removing (-0.5, -0.5) will change the decision boundary becausethe point is a
 support vector. (0.8, 0.2) is not, so removing that point willnot affect the de
cision boundary

e. This would lie on the positive decision boundary, and would certainly affect
it.We would proceed to then use RBF to fit the test data without error.

f. Any value other than 1 will change the hard margin SVM to a soft margin SVM.T
he bigger C is, the 'tighter' the decision boundary would become in order to fit
all of the training samples. The smaller C is, the 'looser' it would get, puttin
gless emphasis on a perfect fit, allowing 'stragglers' to be misclassified. Only
 support vectorsare affected.
```

4. Answer found by running program NN2


luke@ScaryFeet: ~/Dev/classes/csci5521/machine-learning-hw4/csci5521-hw4/hw5
File Edit View Search Terminal Help

```
 [0.9880844 ]
 [0.00205368]
 [0.98656139]
 [0.007943  ]]
 to see convergence, type: plot_error()
luke@ScaryFeet:~/Dev/classes/csci5521/machine-learning-hw4/csci5521-hw4/hw5$ ato
m .
luke@ScaryFeet:~/Dev/classes/csci5521/machine-learning-hw4/csci5521-hw4/hw5$ ato
m .
luke@ScaryFeet:~/Dev/classes/csci5521/machine-learning-hw4/csci5521-hw4/hw5$ pyt
hon3 NN2pruned.py
final coefficients
[[-22.52998438  -9.06859334   9.21414367   1.82615525]
 [-20.32916602   8.1709755  -15.20483478  46.48153248]
 [  0.40645494   1.54775046  -8.8922147   -2.78229687]]
[[ 10.07096998 -11.5603813   -1.83689772]
 [ -6.53343357   7.49518666  -1.31884973]
 [ -4.8437361    3.18934661   0.02161765]
 [  6.13411511  -6.04480838  -2.45174336]]
[[-13.68963598]
 [ 15.41124442]
 [  2.49959651]]

OUTPUTS
[[2.45558814e-03]
 [9.99126258e-01]
 [9.97064558e-01]
 [3.56980591e-04]
 [9.97712894e-01]
 [3.05759847e-03]]
 to see convergence, type: plot_error()
luke@ScaryFeet:~/Dev/classes/csci5521/machine-learning-hw4/csci5521-hw4/hw5$ 
```