M22AI564_Preceptron1

April 29, 2023

Importing All Neccessary Library

[2]:

```
import numpy as np
import matplotlib.pyplot as plt
from prettytable import PrettyTable
```

Converting the given Data Points into Mean Centered Dataset

[3]:

```
x1 = np.array([[1], [-1], [0], [0.1], [0.2],[0.9]])
x2 = np.array([[1], [-1], [0.5], [0.5], [0.2], [0.5]])
print("\n X1 : \n", x1)
print("\n X2 : \n", x2)

data = np.hstack((x1, x2)).tolist()
y = np.array([1, -1, -1, -1, 1, 1])
print("\n class : \n", y)

data = np.hstack((x1, x2)).tolist()
print(data)
# Define column headers
headers = ["x1", "x2", "Class"]
sample_data = PrettyTable(headers)
mean_squared_table = PrettyTable(headers)

for i,row in enumerate(data):
    sample_data.add_row(row+[y[i]])
print("\n Sample Data \n",sample_data)
print(np.mean(data,axis=0))
data = data - np.mean(data,axis=0)
for i,row in enumerate(data):
    mean_squared_table.add_row(row.tolist()+[y[i]])
print("\n Mean Squared  \n",mean_squared_table)
```

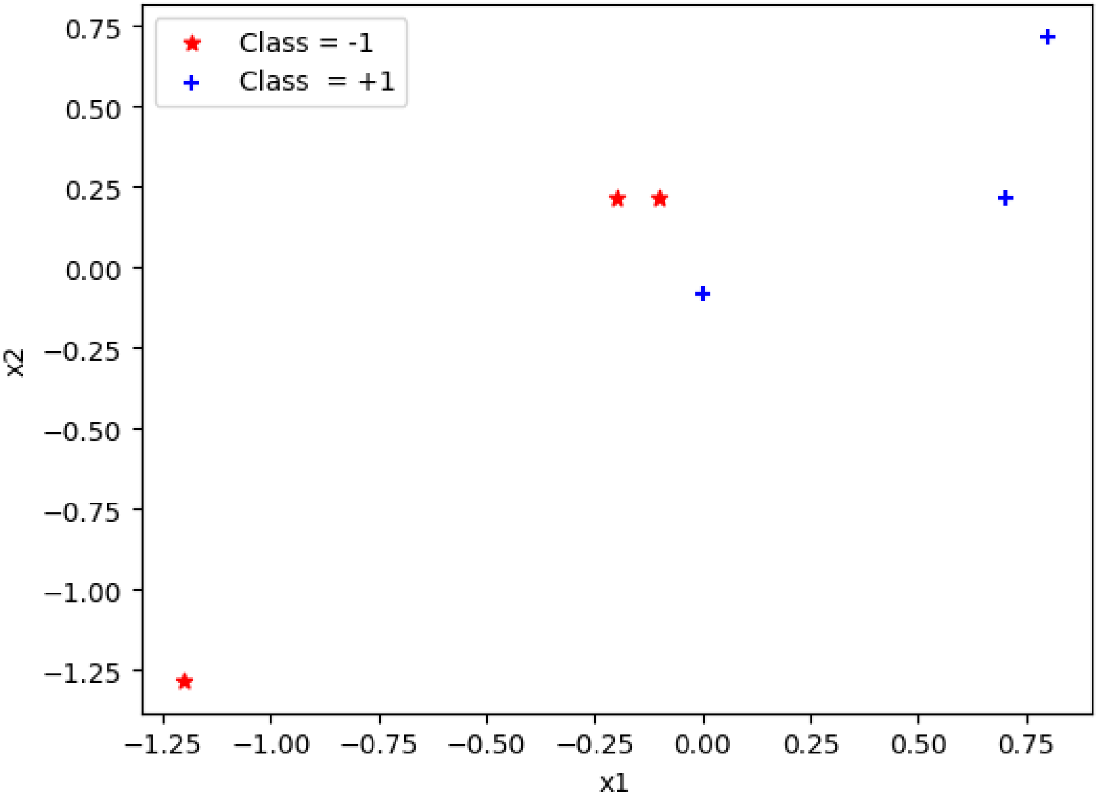
```
X1 :
[[ 1. ]
[-1. ]
[ 0. ]
[ 0.1]
```

```
 [ 0.2]
 [ 0.9]]

 X2 :
 [[ 1. ]
 [-1. ]
 [ 0.5]
 [ 0.5]
 [ 0.2]
 [ 0.5]]

 class :
 [ 1 -1 -1 -1  1  1]
[[1.0, 1.0], [-1.0, -1.0], [0.0, 0.5], [0.1, 0.5], [0.2, 0.2], [0.9, 0.5]]

 Sample Data
 +------+------+-------+
|  x1  |  x2  | Class |
+------+------+-------+
| 1.0  | 1.0  |   1   |
| -1.0 | -1.0 |   -1  |
| 0.0  | 0.5  |   -1  |
| 0.1  | 0.5  |   -1  |
| 0.2  | 0.2  |   1   |
| 0.9  | 0.5  |   1   |
+------+------+-------+
[0.2        0.28333333]

 Mean Squared
 +-------------------------+----------------------+-------+
|            x1           |          x2          | Class |
+-------------------------+----------------------+-------+
|    0.7999999999999999   |  0.7166666666666667  |   1   |
|           -1.2          | -1.2833333333333332  |   -1  |
|   -0.20000000000000004  | 0.21666666666666667  |   -1  |
|   -0.10000000000000003  | 0.21666666666666667  |   -1  |
| -2.7755575615628914e-17 | -0.08333333333333331 |   1   |
|           0.7           | 0.21666666666666667  |   1   |
+-------------------------+----------------------+-------+
```

Plotting the Data Points

[4]:

```python
# Create two arrays for the x and y coordinates of the points
x_coords = data[:,0]
y_coords = data[:,1]
# Loop through the two classes
for i, class_label in enumerate(np.unique(y)):
    # Get the data points for the current class
    class_data = data[y == class_label]
    # Get the x and y coordinates for the current class
    class_x = class_data[:,0]
    class_y = class_data[:,1]
    # Plot the data points with different symbols and colors for each class
    if class_label == 1:
        plt.scatter(class_x, class_y, marker='+', color='blue', label='Class  =␣
 ↪+1')
    else:
        plt.scatter(class_x, class_y, marker='*', color='red', label='Class =␣
 ↪-1')
# Set the axis labels and legend
plt.xlabel('x1')
plt.ylabel('x2')
plt.legend()
# Show the plot
plt.show()
```



Defining Graph Function to plot Graph in each iterations

[9]:

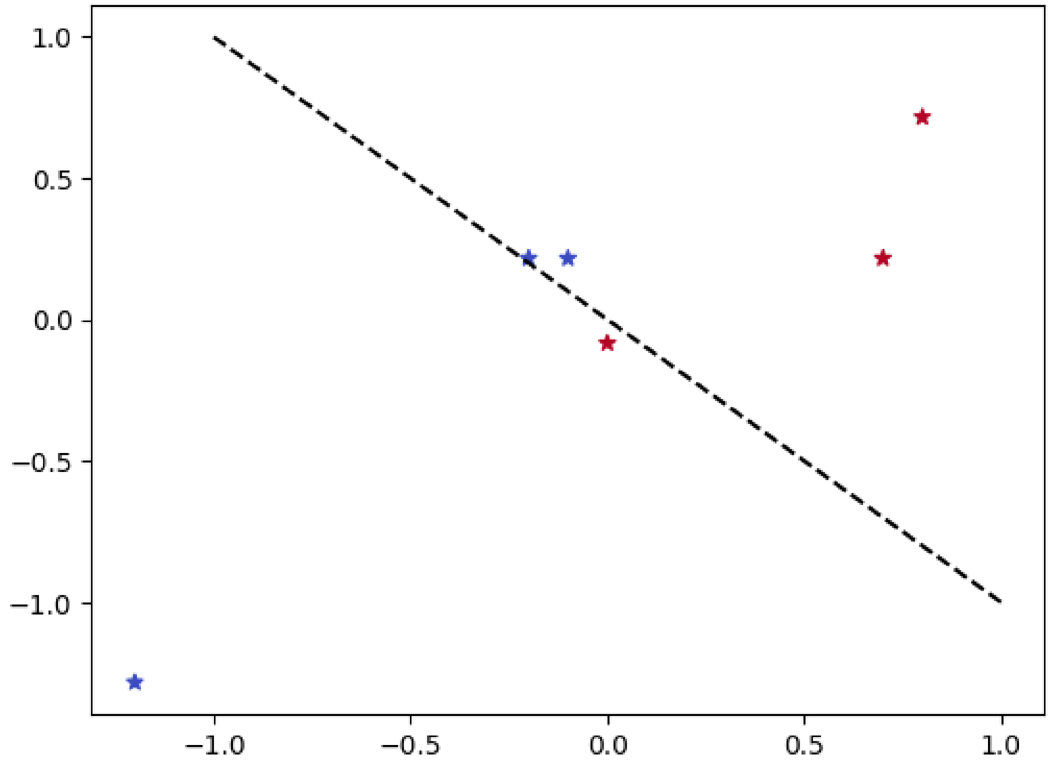
```
def graph(data,w):
    plt.scatter(data[:,0], data[:,1], marker='*',c=y, cmap='coolwarm')
    plt.clf()
    plt.scatter(data[:,0], data[:,1], marker='*',c=y, cmap='coolwarm')
    slope = -w[1] / w[2]
    intercept = -w[0] / w[2]
    x_vals = np.array([-1, 1])
    y_vals = intercept + slope * x_vals
    plt.plot(x_vals, y_vals, '--', color='black')
    plt.show()
```

Preceptron Algorithm

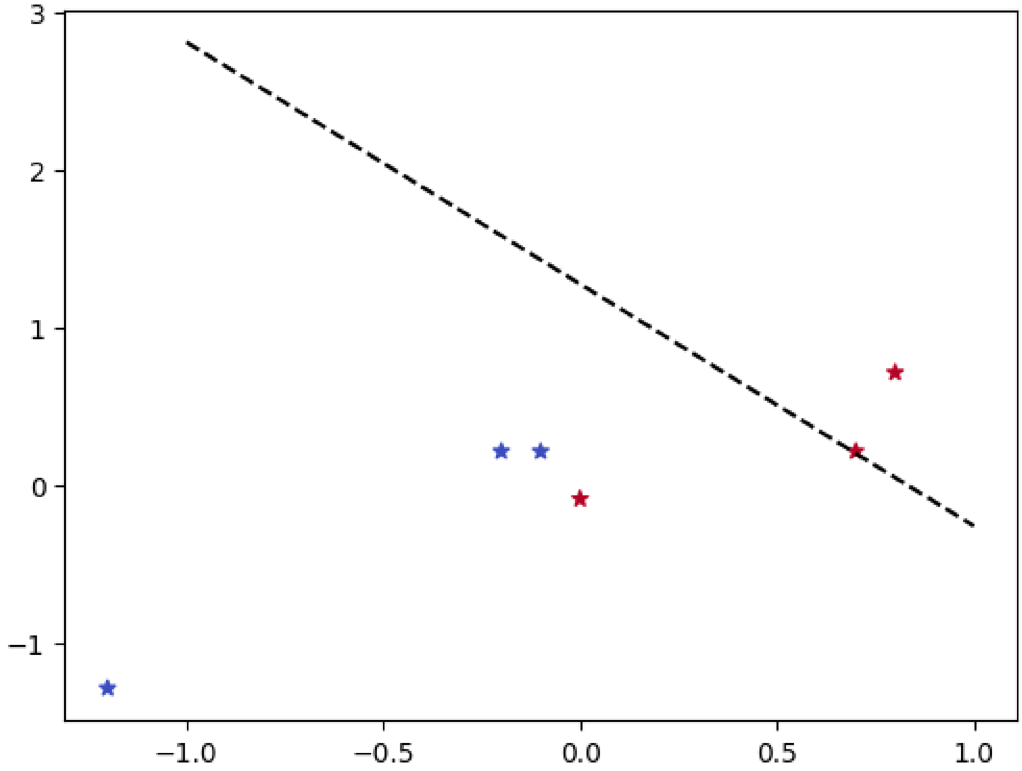
[10]:

```
correctClassified = 0
w=[0,1,1]
graph(data,w)
while (correctClassified != len(data)): #Until everything is classified
  for sample in range(len(data)):
    x = np.append(1,data[sample,0:2])
    print("***Value X****",x)
    if y[sample]==1:
        print(np.dot(np.transpose(w),x))
        if np.dot(np.transpose(w),x)>=0:
            correctClassified=correctClassified+1
            print("sample is pos")
        else: #orange is classified as apple
          w=w+x
          print("******Miss Classified - weight Update*****", "\n",w)
          graph(data,w)
          break

    else:
        print(np.dot(np.transpose(w),x))
        if np.dot(np.transpose(w),x)<0:
            correctClassified=correctClassified+1
            print("sample is neg")
        else:
            w=w-x
            print("******Miss Classified - weight Update*****", "\n",w)
            graph(data,w)

            break

  if(correctClassified != len(data)):
      correctClassified=0
print(w)
```

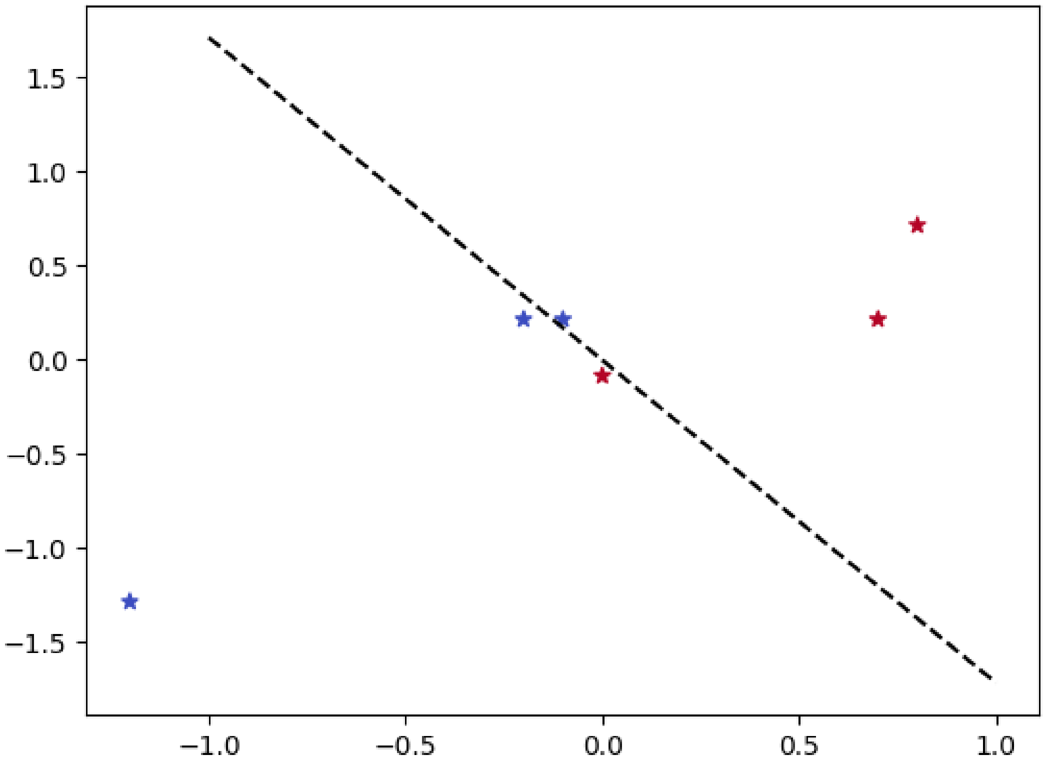
```
# print final weights
print("Final weights: ", w)
```



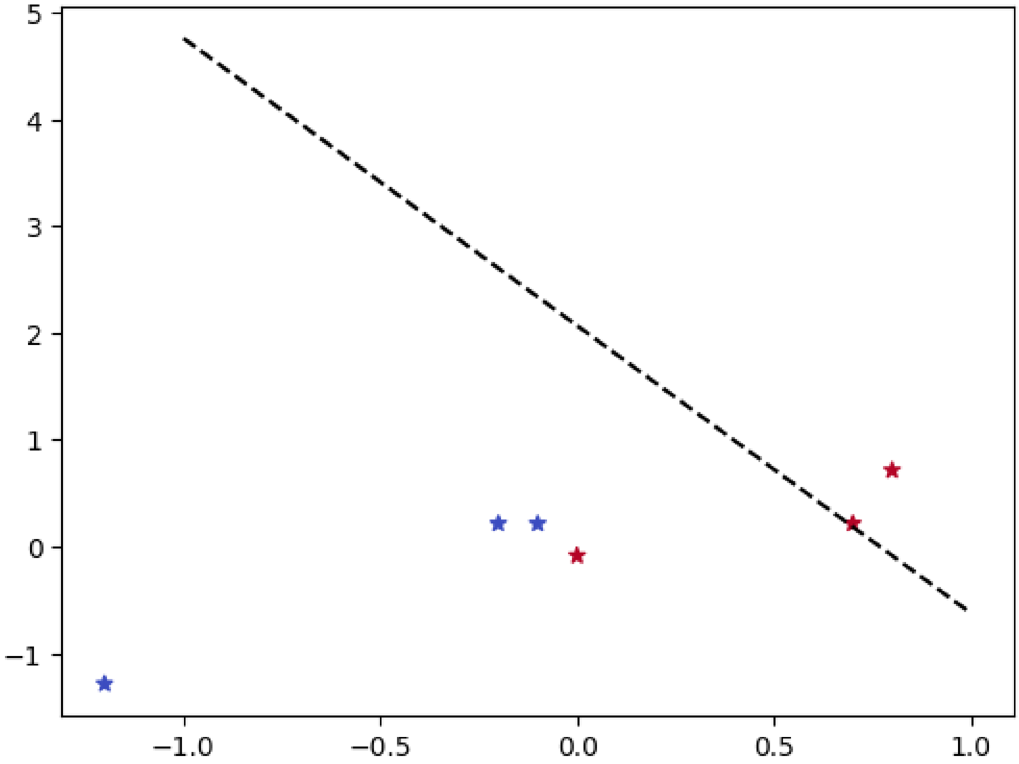
```
***Value X**** [1.         0.8        0.71666667]
1.5166666666666666
sample is pos
***Value X**** [ 1.         -1.2        -1.28333333]
-2.4833333333333334
sample is neg
***Value X**** [ 1.         -0.2         0.21666667]
0.016666666666666635
******Miss Classified - weight Update*****
 [-1.          1.2         0.78333333]
```



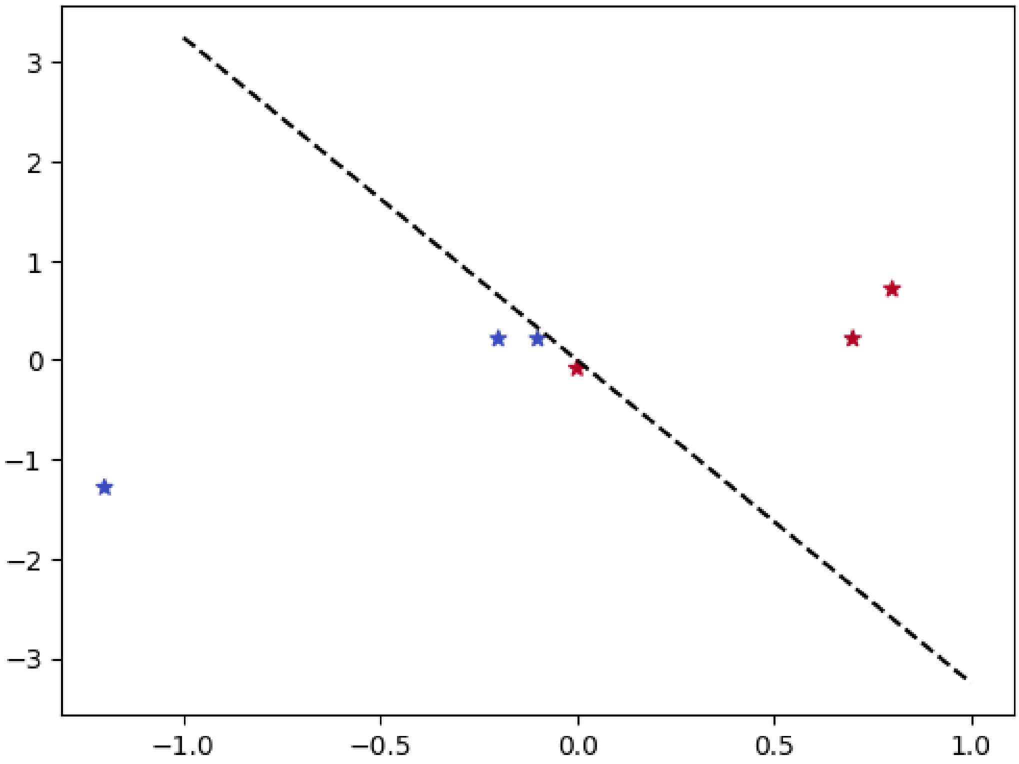
```
***Value X**** [1.         0.8        0.71666667]
0.5213888888888888
sample is pos
***Value X**** [ 1.         -1.2        -1.28333333]
-3.4452777777777777
sample is neg
***Value X**** [ 1.         -0.2         0.21666667]
-1.0702777777777777
sample is neg
***Value X**** [ 1.         -0.1         0.21666667]
-0.9502777777777779
sample is neg
***Value X**** [ 1.00000000e+00 -2.77555756e-17 -8.33333333e-02]
-1.0652777777777778
******Miss Classified - weight Update*****
 [0.  1.2 0.7]
```



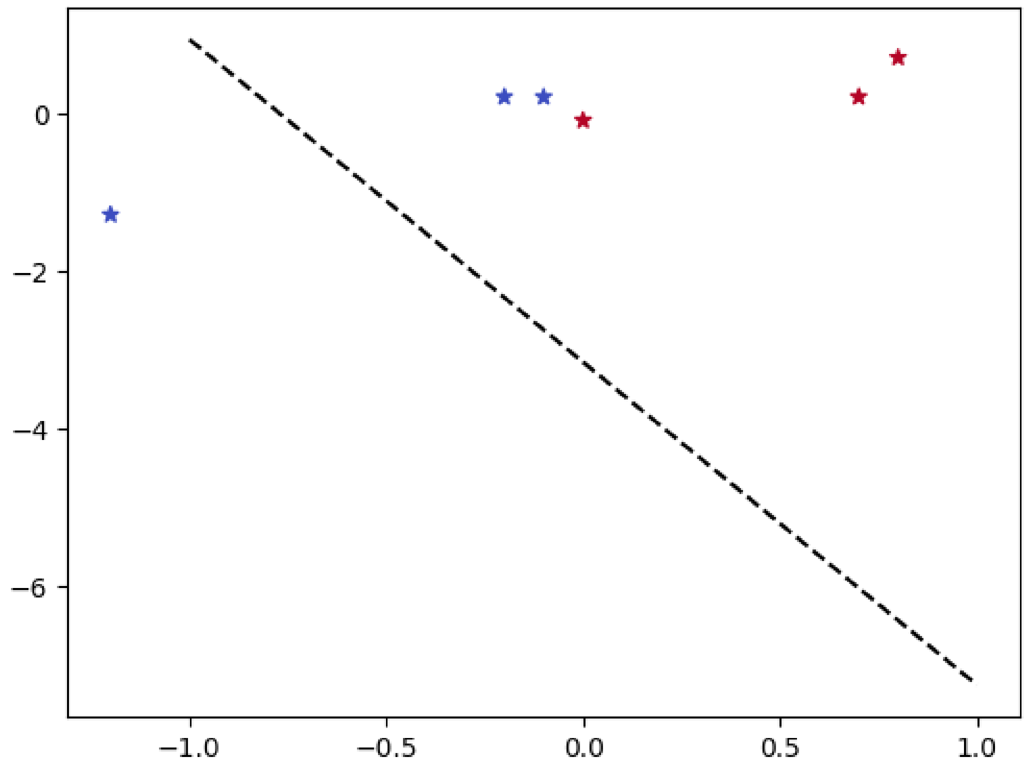
```
***Value X**** [1.         0.8        0.71666667]
1.4616666666666664
sample is pos
***Value X**** [ 1.         -1.2        -1.28333333]
-2.338333333333333
sample is neg
***Value X**** [ 1.         -0.2         0.21666667]
-0.08833333333333337
sample is neg
***Value X**** [ 1.         -0.1         0.21666667]
0.031666666666666635
******Miss Classified - weight Update*****
 [-1.         1.3        0.48333333]
```



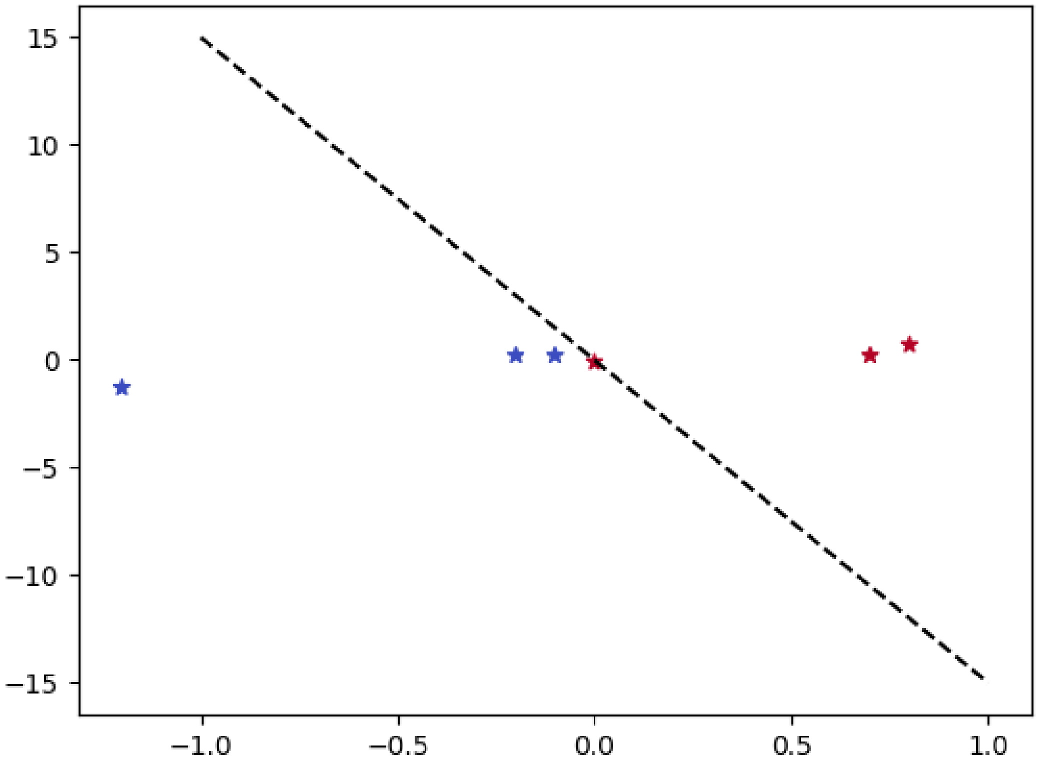
```
***Value X**** [1.          0.8         0.71666667]
0.3863888888888889
sample is pos
***Value X**** [ 1.         -1.2         -1.28333333]
-3.1802777777777775
sample is neg
***Value X**** [ 1.         -0.2          0.21666667]
-1.1552777777777778
sample is neg
***Value X**** [ 1.         -0.1          0.21666667]
-1.0252777777777778
sample is neg
***Value X**** [ 1.00000000e+00 -2.77555756e-17 -8.33333333e-02]
-1.0402777777777779
******Miss Classified - weight Update*****
 [0.  1.3 0.4]
```



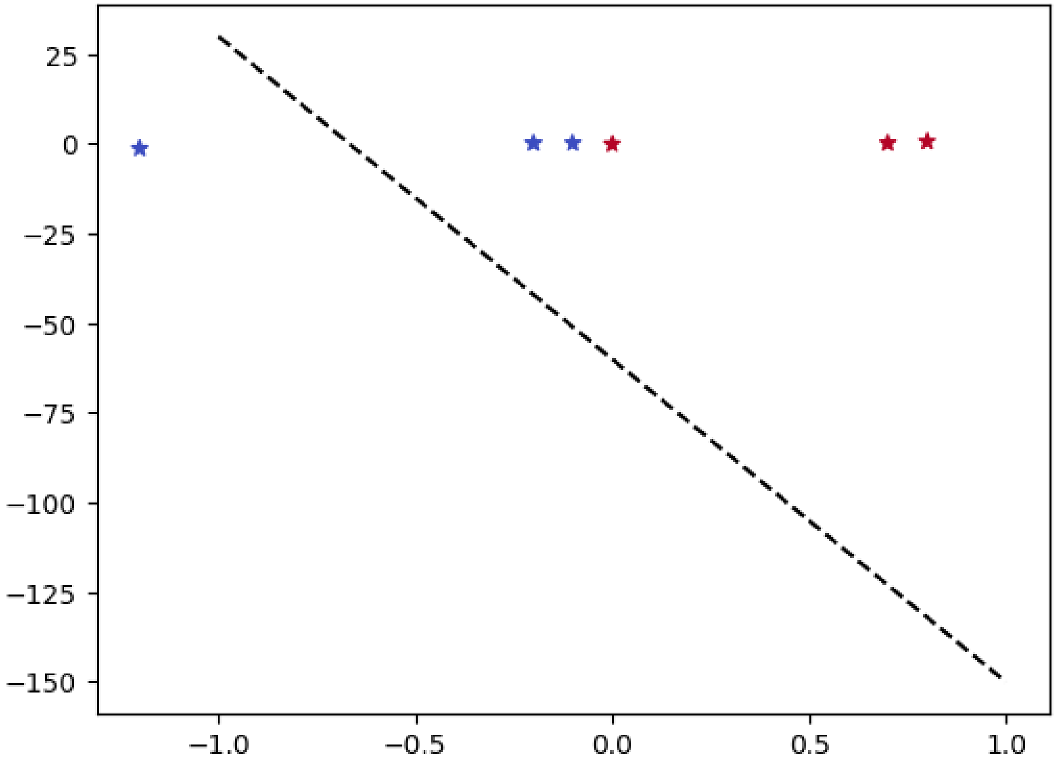
```
***Value X**** [1.          0.8         0.71666667]
1.3266666666666667
sample is pos
***Value X**** [ 1.          -1.2         -1.28333333]
-2.0733333333333333
sample is neg
***Value X**** [ 1.          -0.2          0.21666667]
-0.17333333333333334
sample is neg
***Value X**** [ 1.          -0.1          0.21666667]
-0.043333333333333404
sample is neg
***Value X**** [ 1.00000000e+00 -2.77555756e-17 -8.33333333e-02]
-0.03333333333333336
******Miss Classified - weight Update*****
 [1.         1.3        0.31666667]
```



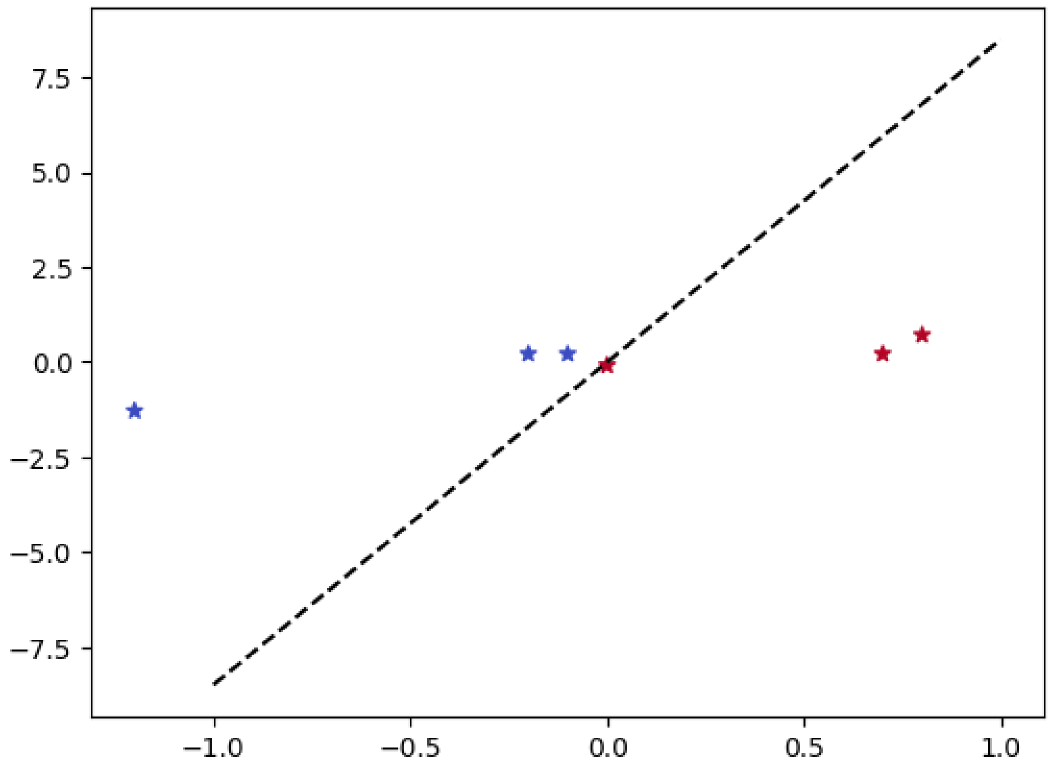
```
***Value X**** [1.          0.8         0.71666667]
2.2669444444444444
sample is pos
***Value X**** [ 1.          -1.2         -1.28333333]
-0.9663888888888889
sample is neg
***Value X**** [ 1.          -0.2          0.21666667]
0.8086111111111111
******Miss Classified - weight Update*****
 [0.  1.5 0.1]
```



```
***Value X**** [1.          0.8         0.71666667]
1.2716666666666665
sample is pos
***Value X**** [ 1.          -1.2         -1.28333333]
-1.9283333333333332
sample is neg
***Value X**** [ 1.          -0.2          0.21666667]
-0.2783333333333334
sample is neg
***Value X**** [ 1.          -0.1          0.21666667]
-0.12833333333333338
sample is neg
***Value X**** [ 1.00000000e+00 -2.77555756e-17 -8.33333333e-02]
-0.008333333333333371
******Miss Classified - weight Update*****
 [1.         1.5        0.01666667]
```



```
***Value X**** [1.          0.8        0.71666667]
2.2119444444444447
sample is pos
***Value X**** [ 1.         -1.2        -1.28333333]
-0.8213888888888887
sample is neg
***Value X**** [ 1.         -0.2         0.21666667]
0.7036111111111111
******Miss Classified - weight Update*****
 [ 0.   1.7 -0.2]
```



```
***Value X**** [1.         0.8        0.71666667]
1.2166666666666666
sample is pos
***Value X**** [ 1.         -1.2        -1.28333333]
-1.7833333333333334
sample is neg
***Value X**** [ 1.         -0.2         0.21666667]
-0.3833333333333334
sample is neg
***Value X**** [ 1.         -0.1         0.21666667]
-0.21333333333333337
sample is neg
***Value X**** [ 1.00000000e+00 -2.77555756e-17 -8.33333333e-02]
0.016666666666666614
sample is pos
***Value X**** [1.         0.7        0.21666667]
1.1466666666666665
sample is pos
[ 0.   1.7 -0.2]
Final weights:  [ 0.   1.7 -0.2]
```