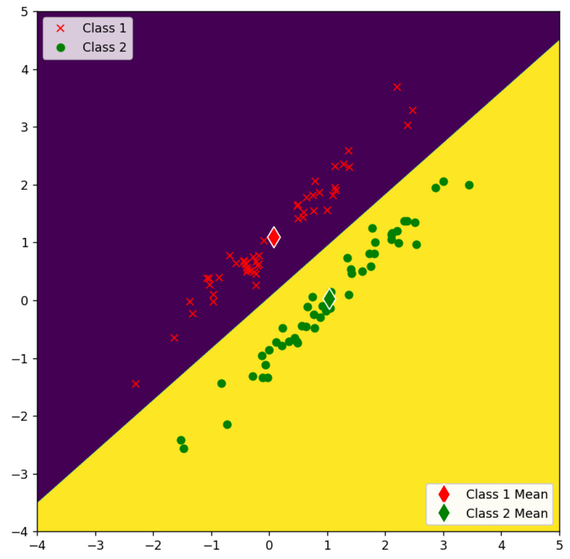
EE 559 **Homework 1 Solutions** Posted: Thursday, 02/02/2023
Jenkins Maximum points – 100

Problem 1. Part (a)

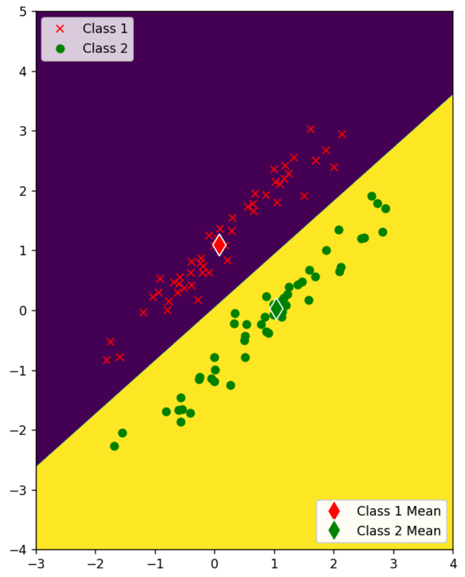
(i) **(12 points)** Correctly plot the training data, class means, decision boundary and decision regions. **4 points** for each dataset. Partial credits are given if some of the parts are missing or are incorrect. No points deducted for not plotting the test set.

Dataset 1

Training data

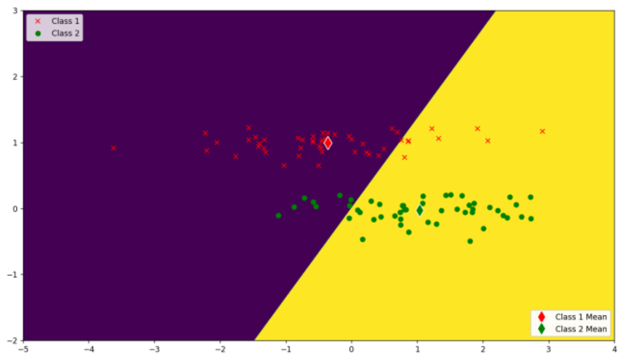


Test data

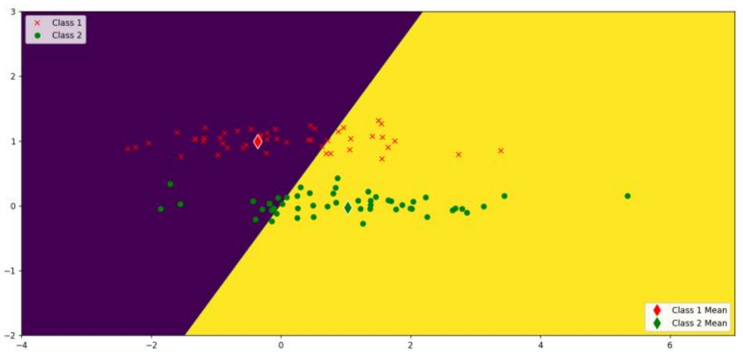


Dataset 2

Training data

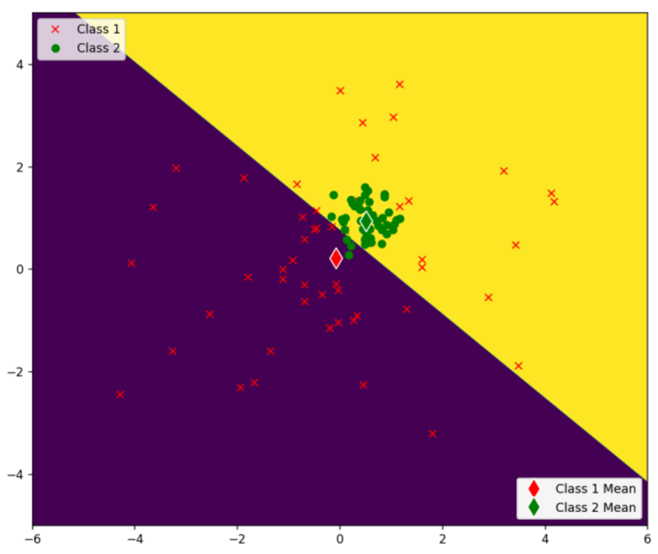


Test data

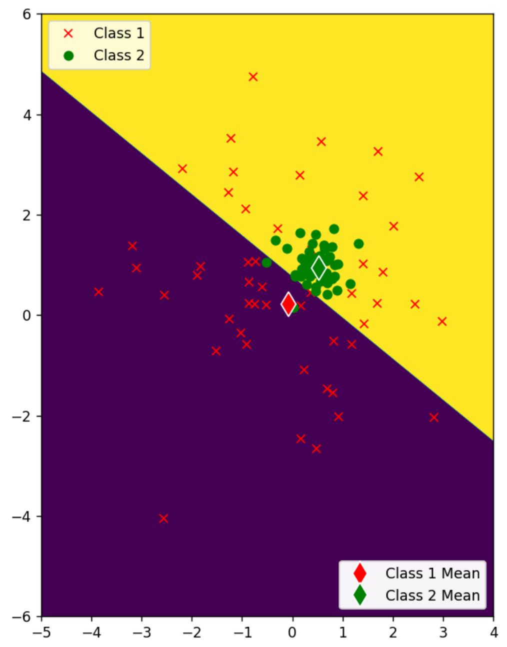


Dataset 3

Training data



Test data



(ii) **(6 points)** Correct classification error rate on training set. **2 points** for each dataset.
(6 points) Correct classification error rate on test set. **2 points** for each dataset.

Dataset	Training error (%)	Test error (%)
1	0	0
2	17	26
3	23	23

Problem 1. Part (b)
(6 points) Reasonable justification of obtained plots and error rate. **2 points** for each dataset.

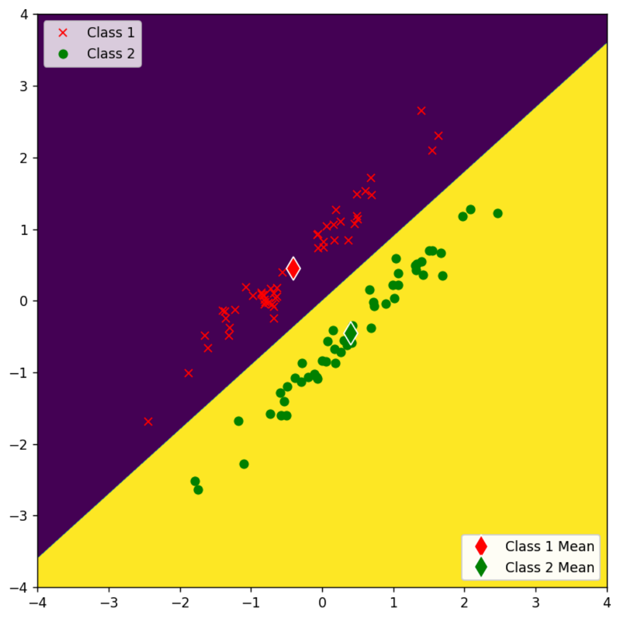
Datasets 1 and 2 are linearly separable, while dataset 3 is not linearly separable. So, ideally, datasets 1 and 2 should be well classified using a linear classifier and we expect a higher error rate for dataset 3 with a linear classifier.
For the nearest means classifier, the class means in dataset 1 are such that the decision boundary (perpendicular to the line joining the two class means) is almost parallel to the direction of maximum variance in the two classes. Hence, the train and test sets are correctly classified.
The class means in dataset 2 are such that the decision boundary is no longer parallel to the direction of maximum variance in the two classes. Hence, some of the data points are misclassified.
Since dataset 3 is not linearly separable, the error rate is high using the nearest means classifier which has a linear decision boundary.

Problem 1. Part (c)

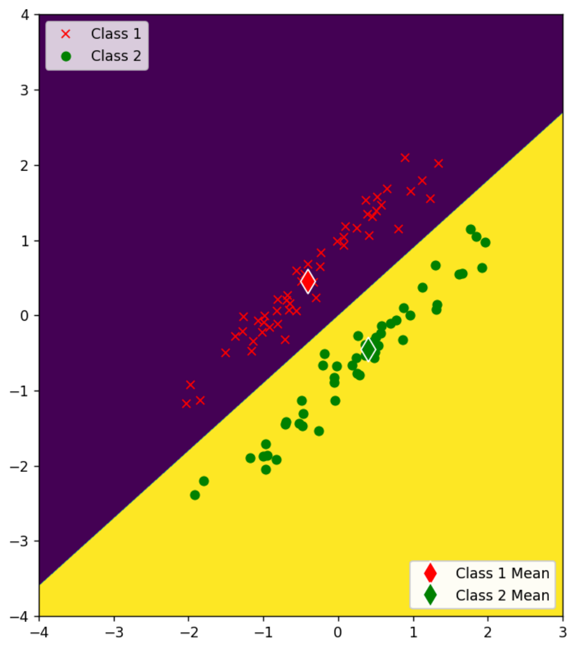
(i.) **(12 points)** Correctly plot the standardized training data, class means, decision boundary and decision regions. **4 points** for each dataset. Partial credits are given if some of the parts are missing or are incorrect. No points deducted for not plotting the test set.

Dataset 1

Training data

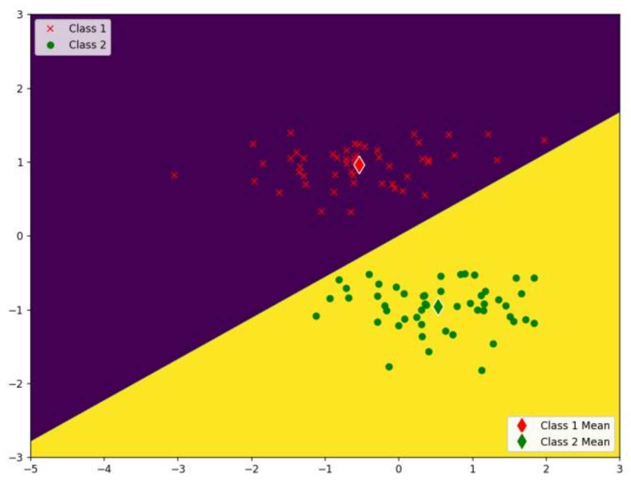


Test data

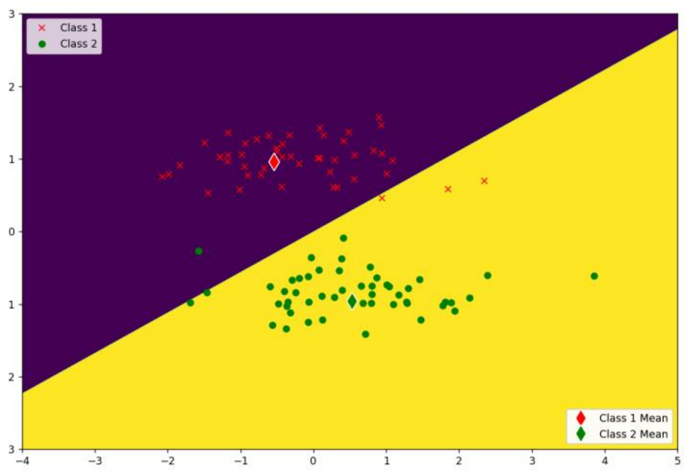


Dataset 2

Training data

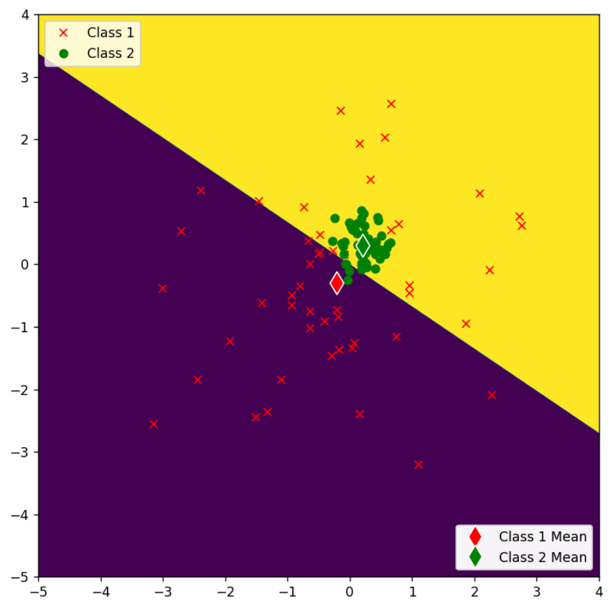


Test data

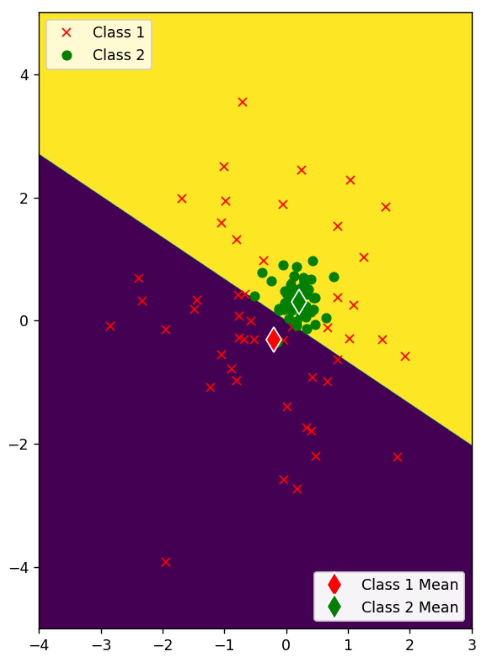


Dataset 3

Training data



Test data



(ii.) **(6 points)** Correct classification error rate on standardized training set. **2 points** for each dataset.
(6 points) Correct classification error rate on standardized test set. **2 points** for each dataset.

Dataset	**Training error (%)**	**Test error (%)**
1	0	0
2	0	4
3	24	21

Problem 1. Part (d)
(6 points) Reasonable justification of obtained plots and error rate. **2 points** for each dataset.

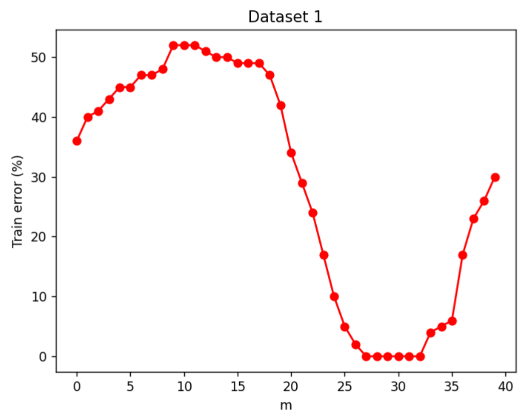
Datasets 1 and 2 remain linearly separable after standardization, while dataset 3 is still not linearly separable.
Standardization does not affect the nature of the decision boundary which is still parallel to the direction of maximum variance in the two classes. So, all the data points are correctly classified similar to part (a).
For dataset 2, due to standardization, the spread in the horizontal direction has reduced (low variance than part (a)). Due to this, most of the misclassified points from part (a) are now correctly classified.

For dataset 3, even after standardization, the means lie very close to each other, and the error rate is similar to part (a) with no standardization.

Problem 1. Part (e)

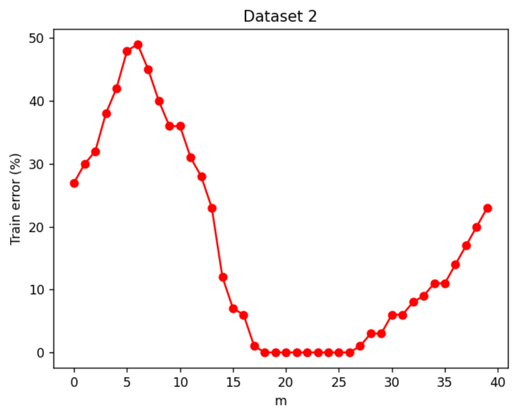
(9 points) Correct plot of m vs. training error rate. **3 points** for each dataset.
(4 points) Report correct r_m^*. **1.5 points** each for datasets 1 and 2. **1 point** for dataset 3. Note that for datasets 1 and 2, there are multiple values of m for which the training error is minimum. You are expected to select the least of these m values.

Dataset 1

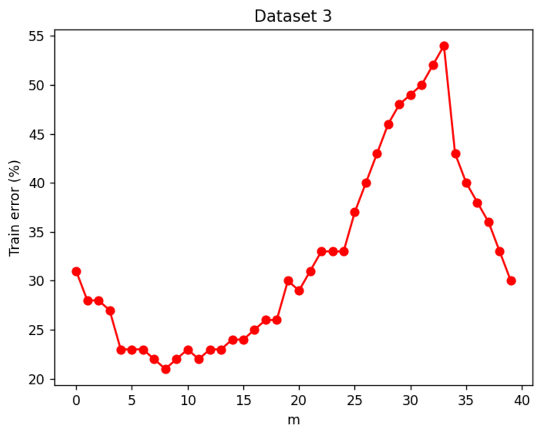


$\boldsymbol{m^*}$ **= 27** $\boldsymbol{r_m^*}$ **= (-7,10)**

Dataset 2



$\boldsymbol{m^*}$ **= 18** $\boldsymbol{r_m^*}$ **= (2,10)**



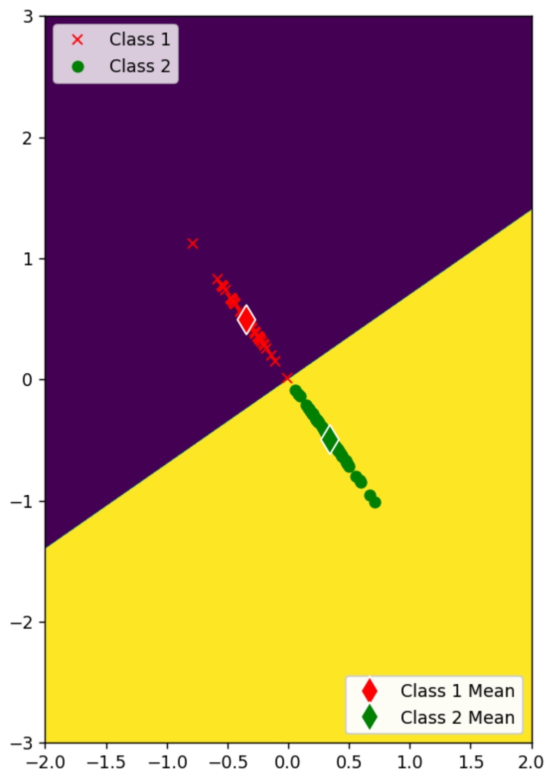
$\boldsymbol{m^*}$ **= 8** $\boldsymbol{r_m^*}$ **= (10,8)**

(12 points) Correct plot of training data, class means, decision boundary and decision regions. **4 points** for each dataset. Since it was not explicitly mentioned to plot the **training** data, full credit would be given if you plot the test set instead. However, the class means still have to be of the training set.

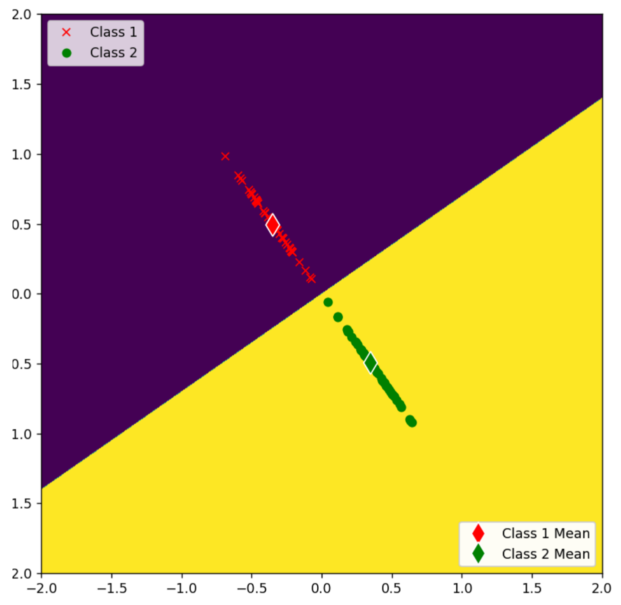
Partial credit will be given if the plotted data is not along the direction of projection, provided the test error rate is correct. (In such case, your plots should look like rotated versions of the following plots.)

Dataset 1

Training data

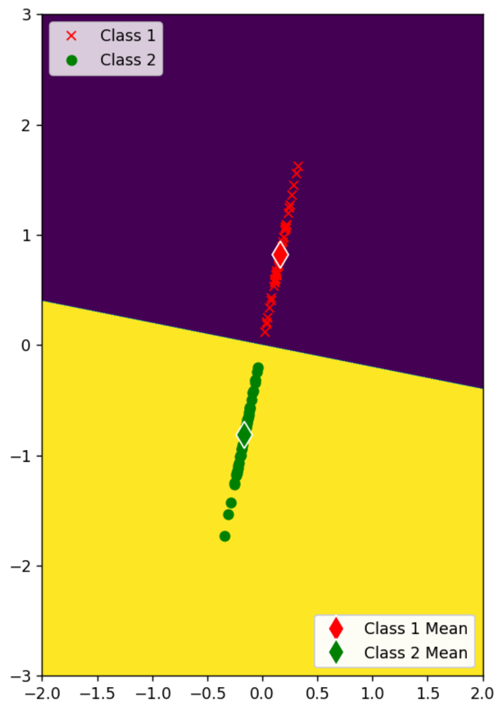


Test data

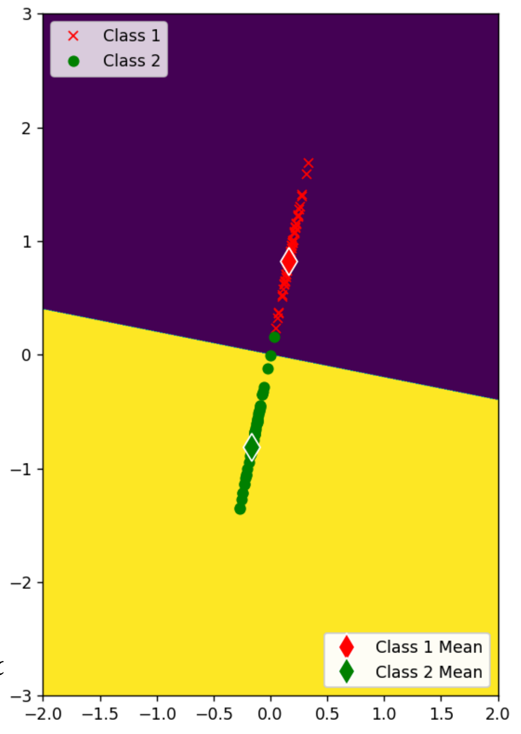


Dataset 2

Training data

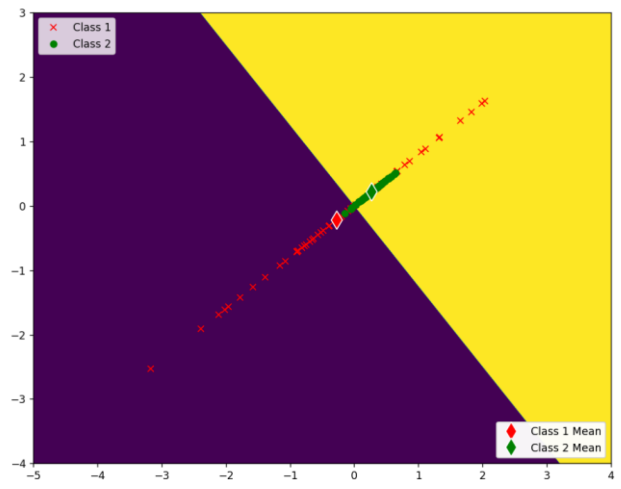


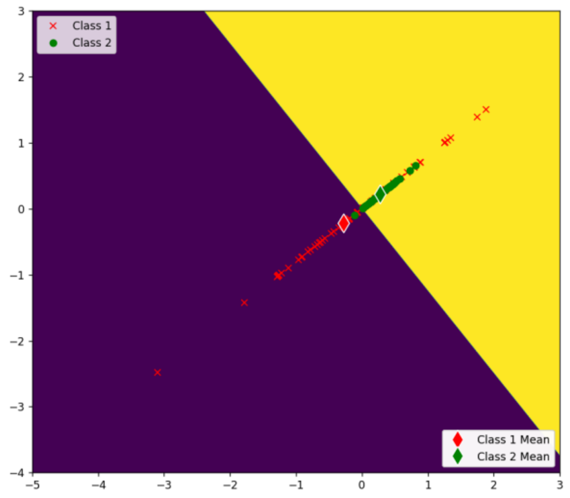
Test data



Dataset 3

Training data **Test data**





(9 points) Correct classification error rate on the test set. **3 points** for each dataset.

Dataset	**Test error (%)**
1	0
2	1
3	24

Problem 1. Part (f)

(6 points) Reasonable comparison and justification. **2 points** for each dataset.

Since datasets 1 and 2 are linearly separable, there exists a linear decision boundary that can successfully classify them. The 40 projection directions span the 2D plane in a uniform manner. So, it is guaranteed that one of these directly would classify all points with zero (or very less) error rate.
Since dataset 1 was classified perfectly in parts (a) and (c), there is no further room for improvement. In other words, the projection method ends up finding a similar decision boundary to the one in parts (a) and (c). This can be visually seen from the plots for dataset 1.
For dataset 2, the projection method further reduces the error rate on test set. Here, the decision boundary is actually very different from parts (a) and (c). However, since there are a set of m values with the same minimum training error, it may happen that one of those m would give the same decision boundary as in parts (a) and (c). It may also be possible to further reduce the test error to zero using one of those m.

For dataset 3, none of the projection directions help separate the two classes in the 1D space. It is seen that even the best direction still has a training error higher than 20%. Subsequently, the test error rate is also high.

Code

We show implementation of basic operations using Python. Detailed code will not be provided.

```python
'''
train_data -> (N,D)
train_label -> (N,)
test_data -> (N,D)
test_label -> (N,)
'''

# Training the Nearest Means Classifier
means = np.zeros((np.unique(train_label).shape[0],train_data.shape[1]))
for class_id in range(means.shape[0]):
    means[class_id] = np.mean(train_data[train_label==class_id+1],axis=0)

# Prediction
dist_train = np.zeros((train_data.shape[0],np.unique(train_label).shape[0]))
for class_id in range(dist_train.shape[1]):
   dist_train[:,class_id] = np.sqrt(np.sum((train_data - means[class_id,:])**2,axis=1))
pred_train = np.argmin(dist_train,axis=1) + 1

dist_test = np.zeros((test_data.shape[0],np.unique(train_label).shape[0]))
for class_id in range(dist_test.shape[1]):
    dist_test[:,class_id] = np.sqrt(np.sum((test_data - means[class_id,:])**2,axis=1))
pred_test = np.argmin(dist_test,axis=1) + 1

# Error calculation
train_error_rate = np.sum(pred_train != train_label) / train_label.shape[0] * 100
test_error_rate = np.sum(pred_test != test_label) / test_label.shape[0] * 100

# Plotting
plotDecBoundaries(train_data,train_label,means)
plotDecBoundaries(test_data,test_label,means)

# Standardization
data_mean = np.mean(train_data, axis=0)
data_std = np.std(train_data, axis=0)
train_data = (train_data - data_mean) / data_std
test_data = (test_data - data_mean) / data_std
```

```python
# Projection (ex. on (10,0)) (1D feature for training and testing)
direction = np.expand_dims(np.asarray([10,0]),axis=1)
train_data_proj_1D = train_data@(direction/np.linalg.norm(direction))
test_data_proj_1D = test_data@(direction/np.linalg.norm(direction))

# Projection (2D projected coordinates for plotting)
train_data_proj_2D = (train_data_proj_1D@direction.T)/np.linalg.norm(direction)
test_data_proj_2D = (test_data_proj_1D@direction.T)/np.linalg.norm(direction)
```