Problem 1

1. Required Procedures

a) hwgennormaldist.m is used for generating random samples

b) hwdisct.m is used for calculating discriminant functions

c) hweuclidean.m is used for calculating euclidean distance

d) hwmahal.m is used for calculating mahalanobis distance

2. Use 1(b) to classify samples

a) Code can be found in hwceproblem1.m

b) Missclassification error is always %66, model always predicts same class.

d)

e)

f)

Problem 2

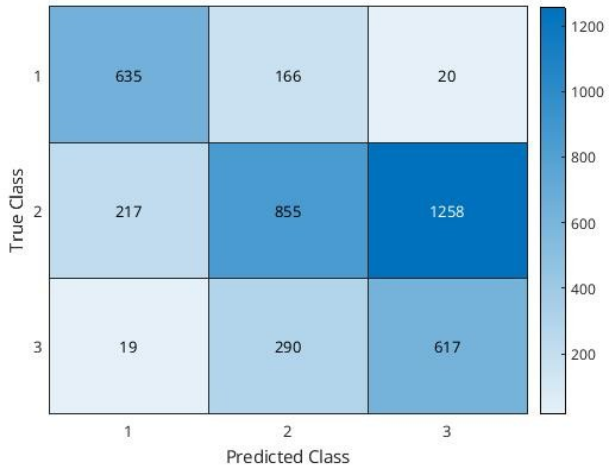
Naïve Bayes Classifier

First 3 features, 100 samples for training, and the rest for test set

Training

Training Loss= %30

Test



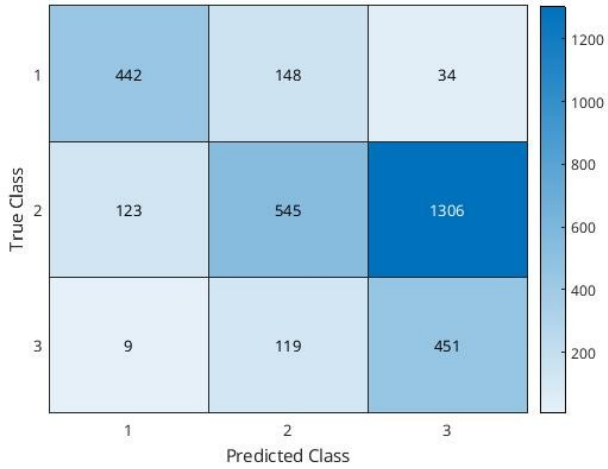
Test Loss = %45.81

First 3 features, 1000 samples for training, and the rest for test set

Training

Training Loss= %36.60

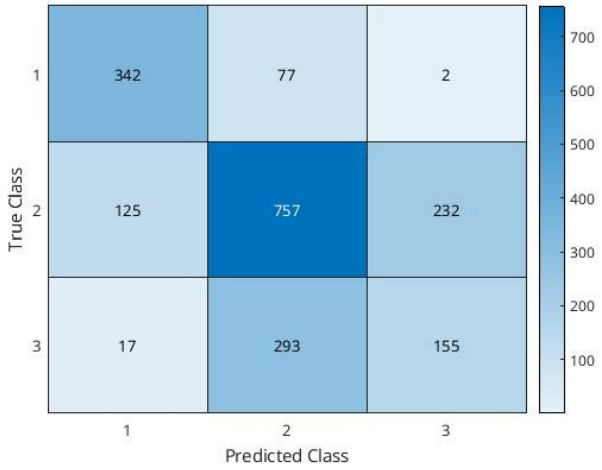
Test



Test Loss = %43.94

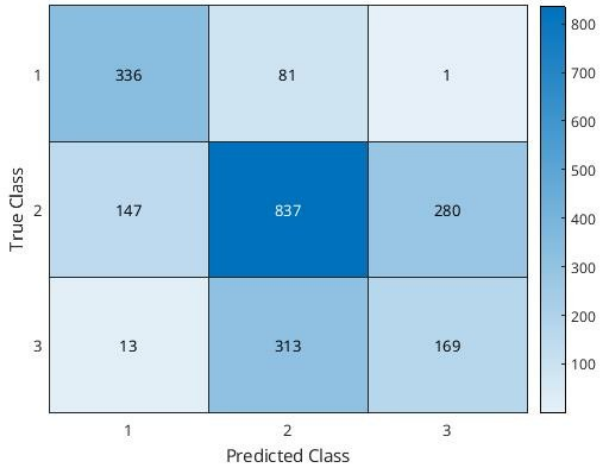
First 3 features, 2000 samples for training, and the rest for test set

Training



Training Loss= %37.30

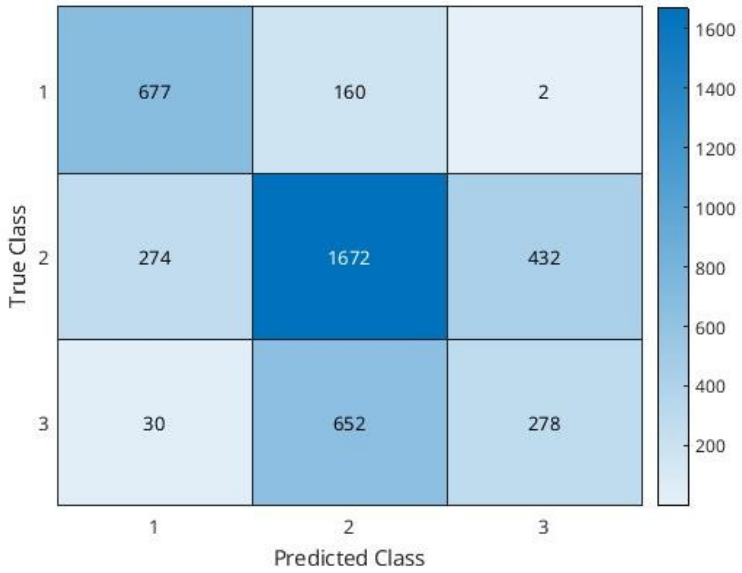
Test



Test Loss = %38.26

First 3 features, 10 fold cross validation

Training



Training Loss = %37.13

All features, 100 samples for training, and the rest for test set

Training

Training Loss= %31

Test

Test Loss = %45.69

All features, 1000 samples for training, and the rest for test set

Training

Training Loss= %37.20

Test

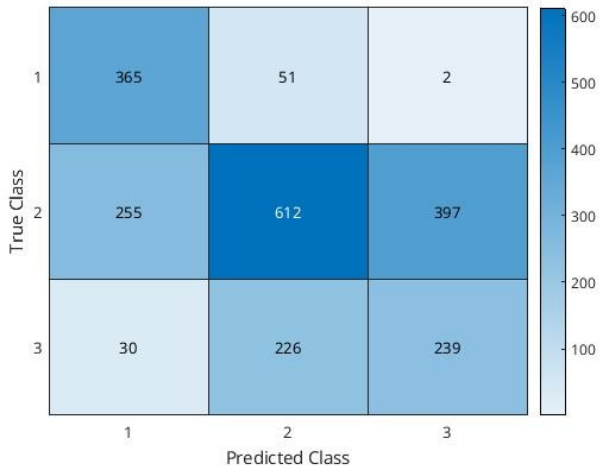
Test Loss = %42.61

All features, 2000 samples for training, and the rest for test set

Training

Training Loss= %41.95

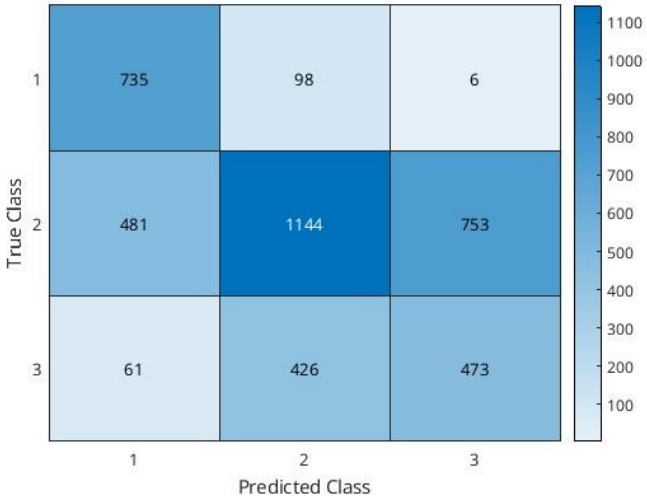
Test



Test Loss = %43.42

All features, 10 fold cross validation

Training



Training Loss = %43.69

Comments

Highest accuracy is obtained at **First 3 features, 10 fold cross validation** case with %37.13 loss**.** It seem for this dataset, using all of the features reduced model's ability to generalise and caused it to make more misclassification errors. On the other hand accuracy has been increased as we increased the training sample, in **First 3 features, 2000 samples for training, and the rest for test set** case model achieved %38.26 loss which is very close to the highest accuracy.

Cross validation divides dataset into k parts, designates one of the parts test set and rest of the parts as training set. Trains k number of models using different combination of training and test sets, compares them and gives us the best performing model. While doing this might be costly in a larger dataset, in our case it allowed us to train a better model than we would have gotten from hand picked train and test datasets.

Code for this problem can be found in hwceproblem2.m.