NORTHEASTERN UNIVERSITY
Department of Electrical and Computer Engineering

EECE 5644 Machine Learning and Pattern Recognition

in

Homework Assignment - 1

Omkar Rajendra Gaikwad
NUID: 002711498
February 20 2023
Github Link for Output and python files

Question 1

The probability density function (pdf) for a 4-dimensional real-valued random vector X is as follows: $p(x) = p(x|L = 0)P(L = 0) + p(x|L = 1)P(L = 1)$. Here L is the true class label that indicates which class-label-conditioned pdf generates the data. The class priors are $P(L = 0) = 0.35$ and $P(L = 1) = 0.65$. The class class-conditional pdfs are $p(x|L = 0) = g(x|m_0, C_0)$ and $p(x|L = 1) = g(x|m_1, C_1)$, where $g(x|m, C)$ is a multivariate Gaussian probability density function with mean vector m and covariance matrix C. The parameters of the class-conditional Gaussian pdfs are:

$$v_0 = \begin{bmatrix} -0.5 \\ -0.5 \\ -0.5 \\ -0.5 \end{bmatrix} \quad C_0 = \frac{1}{4}\begin{bmatrix} 2 & -0.5 & 0.3 & 0 \\ -0.5 & 1 & -0.5 & 0 \\ 0.3 & -0.5 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \quad v_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad C_1 = \begin{bmatrix} 1 & 0.3 & -0.2 & 0 \\ 0.3 & 2 & 0.3 & 0 \\ -0.2 & 0.3 & 1 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

For numerical results requested below, generate 10000 samples according to this data distribution, keep track of the true class labels for each sample. Save the data and use the same data set in all cases.

Part A: : ERM classification using the knowledge of true data pdf:

1. Specify the minimum expected risk classification rule in the form of a likelihood-ratio test: $\frac{p(x|L=1)}{p(x|L=0)} > \gamma$, where the threshold γ is a function of class priors and fixed (non-negative) loss values for each of the four cases $D = i|L = j$ where D is the decision label that is either 0 or 1, like L.

2. Implement this classifier and apply it on the 10K samples you generated. Vary the threshold γ gradually from 0 to ∞, and for each value of the threshold compute the true positive (detection) probability $P(D = 1|L = 1; \gamma)$ and the false positive (false alarm) probability $P(D = 1|L = 0; \gamma)$. Using these paired values, trace/plot an approximation of the ROC curve of the minimum expected risk classifier. Note that at $\gamma = 0$, the ROC curve should be at $(1, 1)$, and as γ increases it should traverse towards $(0, 0)$. Due to the finite number of samples used to estimate probabilities, your ROC curve approximation should reach this destination value for a finite threshold value. Keep track of $P(D = 0|L = 1; \gamma)$ and $P(D = 1|L = 0; \gamma)$ values for each γ value for use in the next section.

3. Determine the threshold value that achieves minimum probability of error, and on the ROC curve, superimpose clearly (using a different color/shape marker) the true positive and false positive values attained by this minimum-P(error) classifier. Calculate and report an estimate of the minimum probability of error that is achievable for this data distribution. Note that $P(\text{error}; \gamma) = P(D = 1|L = 0; \gamma)P(L = 0) + P(D = 0|L = 1; \gamma)P(L = 1)$. How does your empirically selected γ value that minimizes P(error) compare with the theoretically optimal threshold you compute from priors and loss values?

0.0.1 Answer Part A:

1. To begin with this Assignment I generated 10,000 samples from the given probability density Function given in the question, I used 'Numpy' 'Pandas' and 'csv' libraries in 'csv_generator.py' to generate 10,000 samples and store it in a csv file named 'samples_list.csv'. Inorder to generate the samples I used 'multivariate_normal' function from 'scipy' to create a multivariate pdf with two classes viz. L=0, and L=1 for probabilities as P(L=0) = 0.35, and P(L=1) = 0.65. Figure 1 shows the Scatter Plot in 3D of the Samples list.

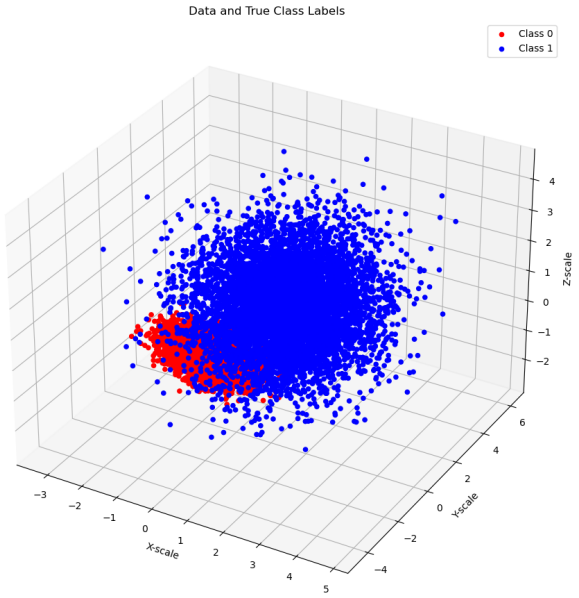

Figure 1: Scatter plot3D of 10,000 samples from a PDF generated.

2. **Calculate Minimum expected risk classification rule**
The minimum classification rule in the form of likelihood-ratio test is:

$$p(x|L=1) = g(x|m1, C1)\ p(x|L=0) \quad = p(x|m0, C0)\ > \gamma = \frac{p(x|L=0)}{p(x|L=1)} \cdot \frac{\lambda_{01} - \lambda_{00}}{\lambda_{10} - \lambda_{11}}$$

The gamma is a threshold function of non negative loss values which are fixed and of Class priors for four cases of D=i— L=j where D is the decision label which has similar value like L (0 or 1). The incorrect results are set to the highest cost possible hence to get correct results we have to go to the lowest cost possible.
We will consider $\lambda_{ij} = \text{Loss}(D = i \,|\, L = j)$, where λ_{ij} is the loss associated with classifying an observation as label i given that the true label was j. The likelihood ratio test used for classifying an observation $\mathbf{x}$ is given by:

$$\frac{p(\mathbf{x} \mid L=1)}{p(\mathbf{x} \mid L=0)} \underset{\text{Decide 0}}{\overset{\text{Decide 1}}{>}} \gamma \triangleq \frac{(\lambda_{10}-\lambda_{00})}{(\lambda_{01}-\lambda_{11})} \frac{p(L=0)}{p(L=1)}.$$

Equating the RHS of the equation as γ and taking the log of both sides, leads to the following decision rule:

$$\ln p(\mathbf{x} \mid L=1) - \ln p(\mathbf{x} \mid L=0) \underset{\text{Decide 0}}{\overset{\text{Decide 1}}{\gtrless}} \ln \gamma.$$

Hence,

$$\gamma = \frac{0.35}{0.65} * \frac{1-0}{1-0} = 0.538$$

Hence,

$$\frac{p(x|L=1}{p(x|L=0)} > 0.538$$

3. Figure 2 shows The ROC Curve generated

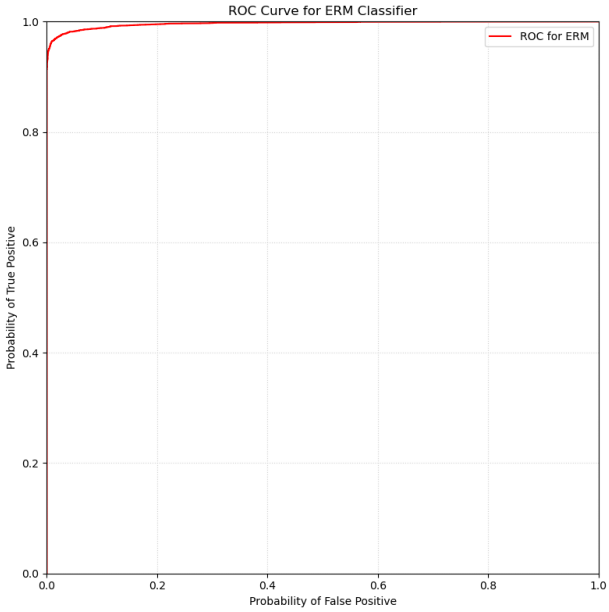

Figure 2: Normal ROC Curve Generated.

4. **Theoretical and Empirical Minimum Risk**
   To calculate the Minimum Risk, I plotted the ROC curve then I found out the list of empirical error probability calculate by a dot product of False positive, and True positive with the ratio of label per sample to the total samples.
   Figure 3 shows the minimum risk roc curve graph for theoretical and empirical error probabilities.

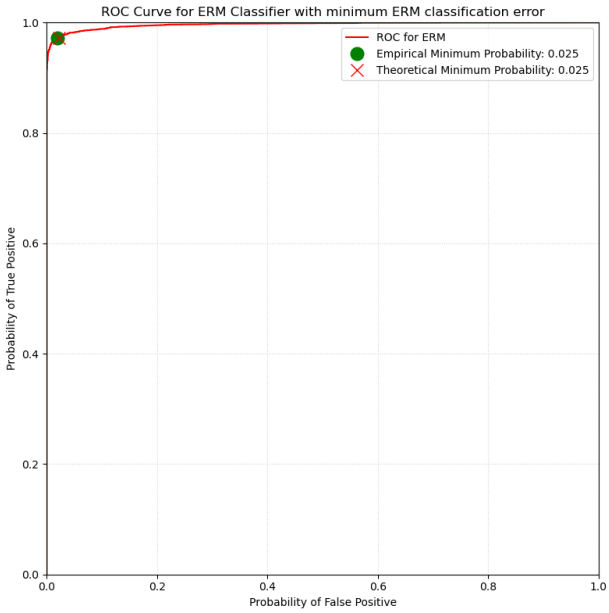

Figure 3: Roc Curve with minimum Risk.

5. Comparing the Theoretical and Experimental Values of Gamma and Probability Error shown in Figure 3
   The Theoritical Gamma = 0.538
   Emperical Gamma = 0.531
   Theoretical Error = 0.025
   Emperical Error = 0.025
   The experimental and theoretical minimum errors are same to conclude they are accurate, We have slight difference in the Gamma.

Part B:

ERM classification attempt using incorrect knowledge of data distribution (Naive Bayesian Classifier, which assumes features are independent given each class label)... For this part, assume that you know the true class prior probabilities, but for some reason you think that the class conditional pdfs are both Gaussian with the true means, but (incorrectly) with covariance matrices that are diagonal (with diagonal entries equal to true variances, off-diagonal entries equal to zeros, consistent with the independent feature assumption of Naive

Bayes). Analyze the impact of this model mismatch in this Naive Bayesian (NB) approach to classifier design by repeating the same steps in Part A on the same 10K sample data set you generated earlier. Report the same results, answer the same questions. Did this model mismatch negatively impact your ROC curve and minimum achievable probability of error?

0.0.2 Answer Part B:

1. To use Naive Bayes classifier to perform binary classification on a dataset. I wrote a program in which the classifier assumes that the features are conditionally independent given the class label, and calculates the likelihood of each class for each data point in the dataset.

2. the code first reads in a dataset from a CSV file, where each row corresponds to a data point and the last column contains the class label (0 or 1). It then calculates the likelihood of each class for each data point using a multivariate normal distribution, where the mean and covariance matrix of the distribution are estimated from the data using a Gaussian Mixture Model (GMM).

3. Next, the code calculates the discriminant score for each data point by taking the log ratio of the likelihoods for class 1 and class 0. This discriminant score is then used to generate a receiver operating characteristic (ROC) curve, which plots the true positive rate (TPR) against the false positive rate (FPR) for different threshold values of the discriminant score. The ROC curve is used to evaluate the performance of the classifier and to compare it to the theoretical minimum error probability.

4. Finally, the code calculates the minimum error probability for the Naive Bayes classifier using both empirical and theoretical methods. The empirical method involves finding the threshold value of the discriminant score that minimizes the error probability on the ROC curve, while the theoretical method involves calculating the minimum error probability based on the class priors and the mean and covariance of the GMM for each class. The code then outputs the minimum error probability and the corresponding threshold value of the discriminant score for the Naive Bayes classifier.

5. Figure 4 Shows the ROc curve of Nauve Bayesian Classifier

6. the procedure to calculate the gamma and error probabilites are same to Part A. The model mismatch did not negatively impact the ROC curve and the probability error are slightly higher.

Part C:

In the third part of this exercise, repeat the same steps as in the previous two cases, but this time using a Fisher Linear Discriminant Analysis (LDA) based classifier. Using the 10K available samples, estimate the class conditional pdf mean and covariance matrices using sample average estimators for mean and covariance. From these estimated mean

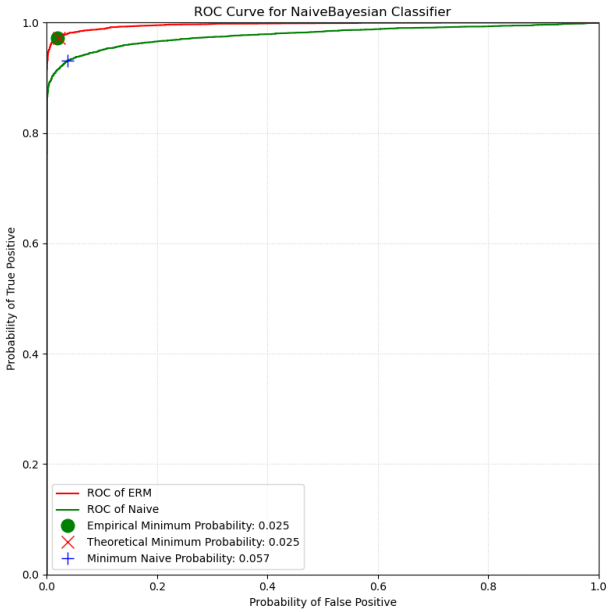

Figure 4: Roc Curve for Naive Bayesian Classifier

vectors and covariance matrices, determine the Fisher LDA projection weight vector (via the generalized eigendecomposition of within and between class scatter matrices): wLDA. For the classification rule w T LDAx compared to a threshold , which takes values from to , trace the ROC curve. Identify the threshold at which the probability of error (based on sample count estimates) is minimized, and clearly mark that operating point on the ROC curve estimate. Discuss how this LDA classifier performs relative to the previous two classifiers.

0.0.3 Answer Part C

1. To Implement LDA Classifier I took following steps:

   (a) I generated the 10K sample data from a mixture of Gaussians specified by the question. Each sample is labeled as 0 or 1 depending on which component of the mixture it was drawn from.

   (b) To estimate roc a vector of LDA discriminant scores and the corresponding true labels is used which gives an output as a dictionary with keys p10 and p11 containing the true positive and false positive data respectively, for different threshold values.

   (c) to perform lda a data matrix X and corresponding labels, to obtain the discriminant score for each sample. It returns the weight vector w used to compute the discriminant score, and the projected data z.

   (d) The code generates data from a mixture of Gaussians, performs LDA on the data, estimates the ROC curve using the estimate_roc function, and plots the ROC curve using matplotlib.

2. Figure 5 shows the LDA Classifier Minimum Probability which is higher than the ERM classification but less than the Naive Bayesian Classifier, SO we can say that Naive Bayesian Classifier gives us less accuracy, second is a Fisher Linear Discriminant Analysis (LDA) based classifier, and the last the most better results are shown by ERM classification.

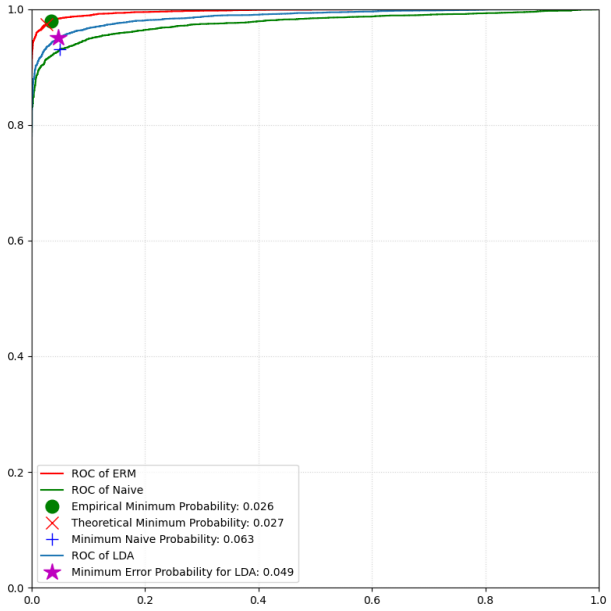

Figure 5: Roc Curve for LDA Classifier

1 Question 2:

A 3-dimensional random vector X takes values from a mixture of four Gaussians. One of these Gaussians represent the class-conditional pdf for class 1, and another Gaussian represents the classconditional pdf for class 2. Class 3 data originates from a mixture of the remaining 2 Gaussian components with equal weights. For this setting where labels L 1,2,3, pick your own class conditional pdfs p(x—L = j), j 1,2,3 as described. Try to approximately set the distances between means of pairs of Gaussians to approximately 2 to 3 times the average standard deviation of the Gaussian compoenents, so that there is some significant overlap between class-conditional pdfs. Set class priors to 0.3,0.3,0.4.

1.1 Part A:

Minimum probability of error classification (0-1 loss, also referred to as Bayes Decision rule or MAP classifier).

1. Generate 10000 samples from this data distribution and keep track of the true labels of each sample.

2. Specify the decision rule that achieves minimum probability of error (i.e., use 0-1 loss), implement this classifier with the true data distribution knowledge, classify the 10K samples and count the samples corresponding to each decision-label pair to empirically estimate the confusion matrix whose entries are P(D = i—L = j) for i, j 1,2,3.

3. Provide a visualization of the data (scatter-plot in 3-dimensional space), and for each sample indicate the true class label with a different marker shape (dot, circle, triangle, square) and whether it was correctly (green) or incorrectly (red) classified with a different marker color as indicated in parentheses.

1.1.1 Answer Part A

1. Figure 6 shows the 3D scatter plot of 10000 samples generated for the Minimum Probability Error Classification

Figure 6: Sample Scatter Plot in 3D with three classes

2. For a given $\mathbf{x}$, we want to choose a class label i which minimizes risk (or loss) associated with choosing this class label. We know that the ERM decision rule for this problem is based on conditional risk:

$$D(\mathbf{x}) = \underset{i\in\{1,2,3\}}{\operatorname{argmin}} R(D = i \,|\, \mathbf{x}) = \underset{i\in\{1,2,3\}}{\operatorname{argmin}} \sum_{j=1}^{3} \lambda_{ij} p(\mathbf{x} \,|\, L = j) p(L = j),$$

where the expression expands the class posteriors $p(L = j \mid \mathbf{x})$ for $j \in \{1, 2, 3\}$ using Bayes rule.

3. Figure 7 shows the Confusion Matrix

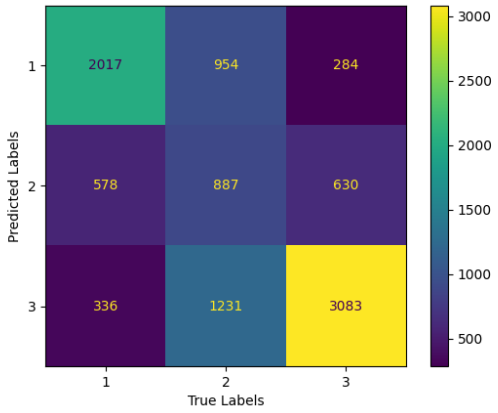

Figure 7: Confusion Matrix for Part A

4. Figure 8 shows the scatter plot of the data in 3D

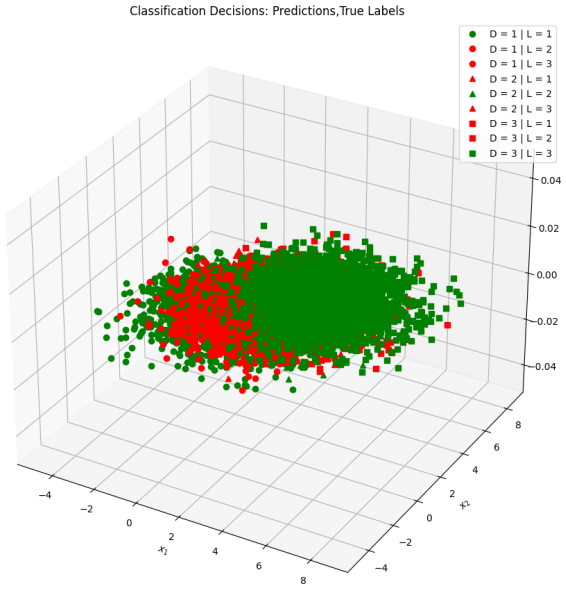

Figure 8: Scatter Plot for Predictions

5. Figure 9 shows the Output of the Part A with Number of Misclassified Samples andEstimated Probability Error

Figure 9: Output for Part A

1.2 Part B:

Repeat the exercise for the ERM classification rule with the following loss matrices which respectively care 10 times or 100 times more about not making mistakes when L = 3

$$\mathbf{\Lambda}10 = \begin{bmatrix} 0 & 1 & 10 \\ 1 & 0 & 10 \\ 1 & 1 & 10 \end{bmatrix} and \mathbf{\Lambda}100 = \begin{bmatrix} 0 & 1 & 100 \\ 1 & 0 & 100 \\ 1 & 1 & 10 \end{bmatrix}$$

Note that, the (i, j)th entry of the loss matrix indicates the loss incurred by deciding on class i when the true label is j. For this part, using the 10K samples, estimate the minimum expected risk that this optimal ERM classification rule will achieve. Present your results with visual and numerical reprentations. Briefly discuss interesting insights, if any.

1.2.1 Answer Part B

1. For Lambda-10 Output are as Follows

   (a) Figure 10 shows Confusion Matrix for loss matrices 10 times

   (b) Figure 11 Scatter plot of the data

2. For Lambda-100 Output are as Follows

   (a) Figure 12 Confusion Matrix for loss matrices 100 times

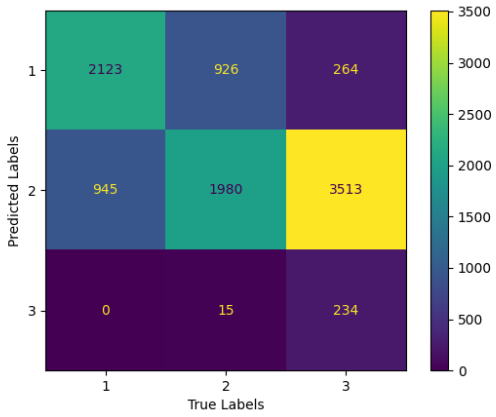

Figure 10: Confusion Matrix for Part B for loss matrices care 10 times

(b) Figure 13 shows the Scatter plot of the data

3. Figure 14 shows Output for Misclassified and Probability error samples

4. The ERM classifier in Part B, which uses a loss matrix, penalizes misclassifications for distant classes more heavily than for adjacent classes, unlike the classifier in Part A. As a result, the Part B classifier may have a higher error rate, as it places more emphasis on correctly classifying outlier samples that could be mistaken for distant, incorrect classes. However, this may result in a lower penalty for errors on frequently occurring adjacent classes compared to the ¡b¿Part A¡/b¿ classifier. In particular, we expect that Part B's classifier will have lower false probability scores than Part A's classifier in situations where there is a higher risk associated with misclassifying a sample as a distant class, as shown in the confusion matrices. This effect is more noticeable when the class-conditional probability density functions overlap significantly. we can also say that the Error for 100 times is more than that of 10 times.

2 Question 3

Download the following datasets... • Wine Quality dataset located at https://archive.ics.uci.edu/ml/dataset Wine+Quality consists of 11 features, and class labels from 0 to 10 indicating wine quality scores. There are 4898 samples. • Human Activity Recognition dataset located at https://archive.ics.uci.edu/ ml/datasets/Human+Activity+Recognition+Using+Smartphones consists of 561 features, and 6 activity labels. There are 10299 samples. Implement minimum-probability-of-error classifiers for these problems, assuming that the class conditional pdf of features for each class you encounter in these examples is a Gaussian. Using all available samples from a class, with sample averages, estimate mean vectors and covariance matrices.

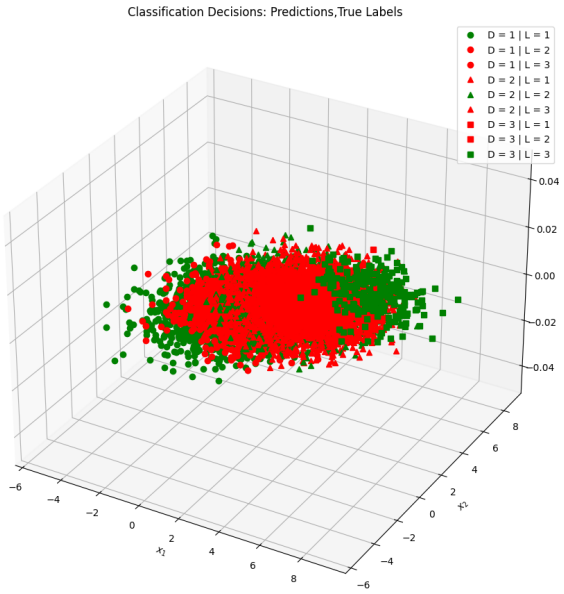

Figure 11: Scatter Plot for Part B for loss matrices care 10 times

Using sample counts, also estimate class priors. In case your sample estimates of covariance matrices are ill-conditioned, consider adding a regularization term to your covariance estimate as in: CRegularized = CSampleAverage +I where ¿ 0 is a small regularization parameter that ensures the regularized covariance matrix CRegularized has all eigenvalues larger than this parameter. With these estimated (trained) Gaussian class conditional pdfs and class priors, apply the minimum-P(error) classification rule on all (training) samples, count the errors, and report the error probability estimate you obtain for each problem. Also report the confusion matrices for both datasets, for this classification rule. Visualize the datasets in various 2 or 3 dimensional projections (either subsets of features, or using the first few principal components). Discuss if Gaussian class conditional models are appropriate for these datasets and how your model choice might have influenced the confusion matrix and probability of error values you obtained in the experiments conducted above. Make sure you explain in rigorous detail what your modeling assumptions are, how you estimated/selected necessary parameters for your model and classification rule, and describe your analyses in mathematical terms supplemented by numerical and visual results in a way that conveys your understanding of what you have accomplished and demonstrated.

2.1 Part A: Wine Dataset

1. The white wine dataset consists of 11 features and class labels from 0 to 10 indicating the scores of wine quality the total number of samples are 4898.

2. The python file 'wine.py does the following steps in detail

   (a) The code is implementing a Gaussian Mixture Model (GMM) for a wine quality dataset.

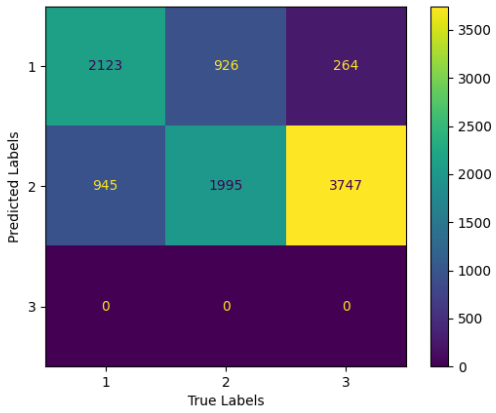

Figure 12: Confusion Matrix for Part B for loss matrices care 100 times

(b) The code starts by defining a function regularized covariance which takes a data matrix data array and regularization parameter lambda, and returns a regularized covariance matrix. The regularization term is added to the diagonal of the covariance matrix to prevent it from being singular. The parameter lambda is usually chosen through cross-validation.

(c) The code then loads the wine quality dataset and extracts the input features X and target labels qualities. A label encoder is defined to encode the labels as 0, 1, ..., C, where C is the number of classes.

(d) Next, the class priors priors are estimated by computing the fraction of samples in each class. The number of classes is inferred from the length of priors.

(e) The mean mu and covariance Sigma of each class are estimated using the maximum likelihood method. The mean is computed by taking the mean of each feature for each class, while the covariance is computed by taking the regularized covariance matrix of the feature matrix for each class. The regularization parameter lambd is set to 1/n, where n is the number of features. This choice of regularization helps to prevent overfitting.

(f) If the loss function is 0-1 loss, then the decision rule is the maximum a posteriori(MAP) decision rule. Otherwise, it is the empirical risk minimization (ERM) classifier, which minimizes the expected loss on the training data.

(g) The code initializes the loss matrix Lambda as a matrix of ones with zeros on the diagonal, which corresponds to 0-1 loss. The MAP decision rule is computed by selecting the label associated with the maximum posterior probability for each observation. The ERM classifier is computed by selecting the label associated with the minimum conditional risk for each observation.

(h) The code then computes the confusion matrix for the decisions and prints it to the console. The diagonal of the confusion matrix represents the number of correctly classified samples for each class, while the off-diagonal entries represent the misclassifications. The total number of misclassified samples and the empirically

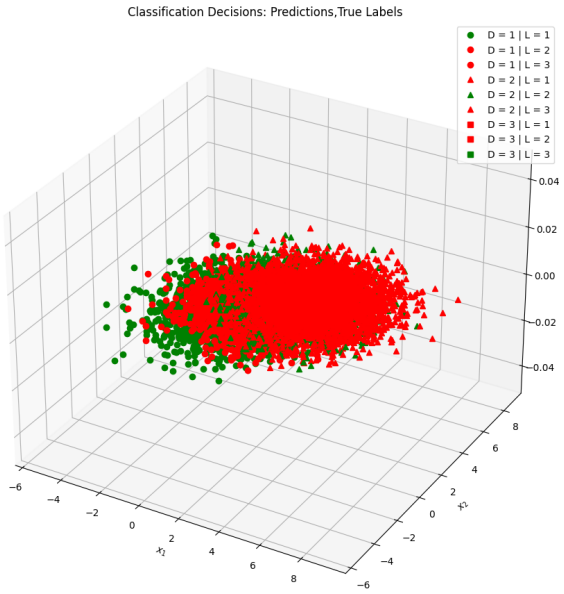

Figure 13: Scatter Plot for Part B for loss matrices care 100 times

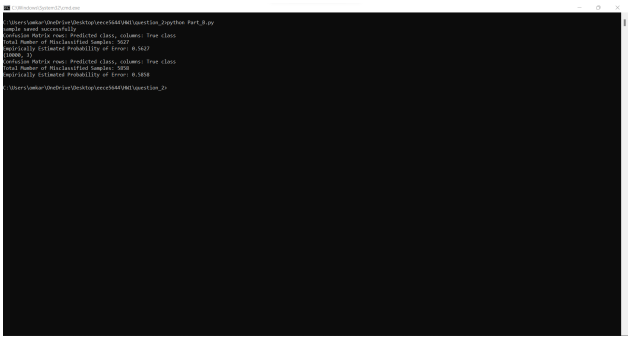

Figure 14: Output for Part B

estimated probability of error are also printed to the console. The probability of error is the proportion of misclassified samples to the total number of samples.

(i) The data is then fitted and reduced to 7 components to find the PCA.

3. Reason of PCA:

   (a) The reason of using PCA is to find the resemblance of the data with Guassian data.

   (b) To find whether we have selected correct Model for the task we are reducing the dimensions using PCA.

4. Figure 15 shows the Confusion Matrix of Wine Database to detect the Misclassified Samples

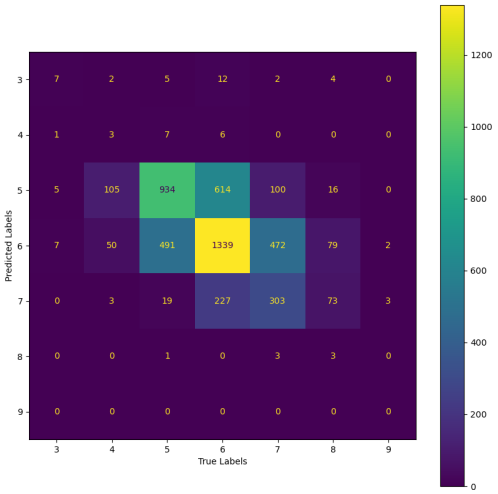

Figure 15: Confusion Matrix of Wine Database to detect the Misclassified Samples

5. Figure 16 shows the Scatter Plot of the subset of Wine database

6. Figure 17 shows the PCA over Wine Database Scatter plot

7. Figure 18 shows the PCA variance ration, and the Empirically Estimated Probability of Error

2.2 Part B: Har Dataset

1. Human Activity Recognition dataset consists of 561 features and 6 activity labels and There are 10299 samples.

2. I used urlopen, ZipFile, and BytesIO to open the training and testing data.

3. To implement PCA over the Har Dataset I reduced 3 components and used a fitted estimator not the actual data to plot the scatter plot.

4. Figure 19 shows the scatter Plot for PCA over Har Dataset

5. Figure 20 shows the PCA variance ratio of the Har dataset

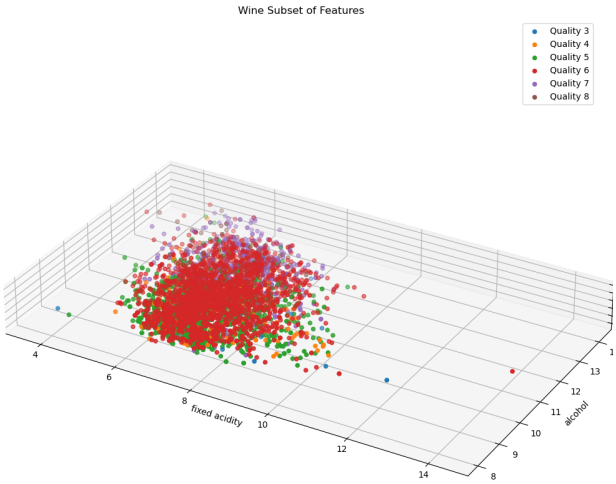

Figure 16: Scatter Plot of the subset of Wine database

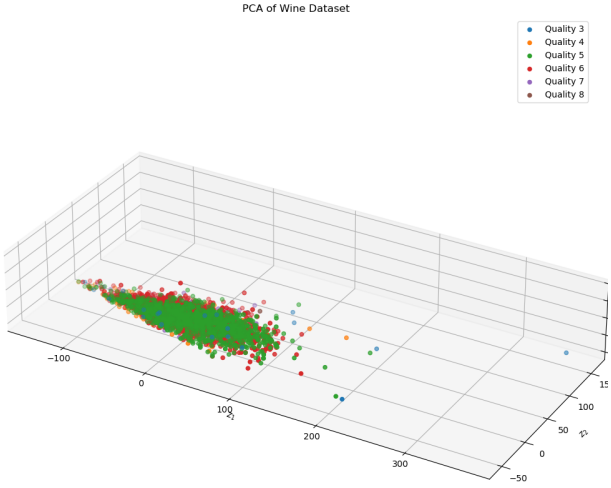

Figure 17: PCA Scatter Plot of the subset of Wine database

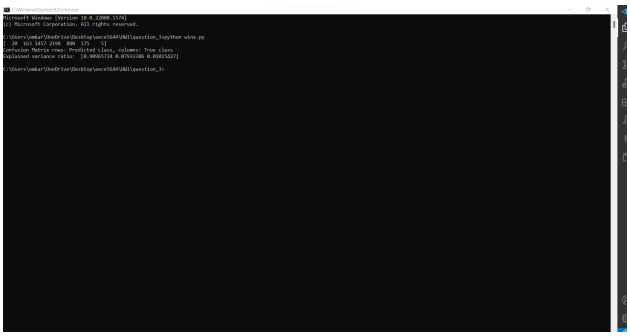

Figure 18: Empirically Estimated Probability Error Output

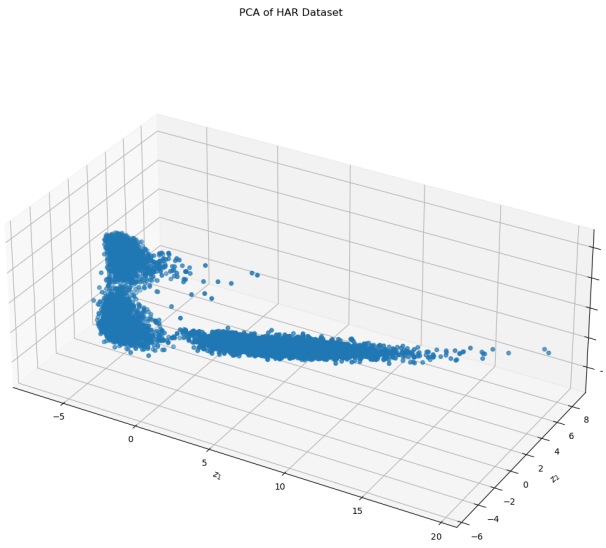

Figure 19: Empirically Estimated Probability Error Output

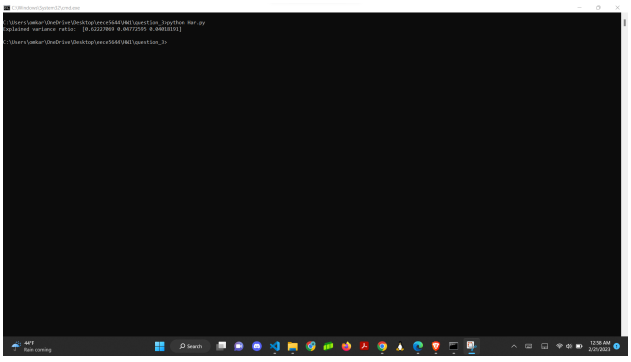

Figure 20: Empirically Estimated Probability Error Output