EECE 5644 Homework 1

Emily Costa (costa.em@northeastern.edu)

February 22, 2021

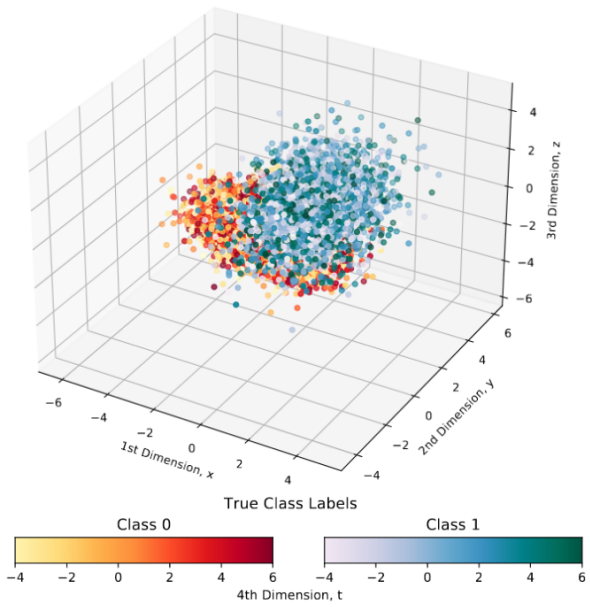

Figure 1: *10,000 sample values from the provided multivariate Gaussian probability density function distribution with two true classes. As higher dimensions are difficult to visualize, I use a gradient colorbar to indicate the value of the fourth dimension.*

1

First, I generated 10,000 samples from the provided probability density function (PDF) provided in the homework. In order to randomly generate these samples, I used the random library from the NumPy API in Python. I then decided how many samples to get from each class by setting the probability of getting a sample from Class 0 to 70% and Class 1 to 30%, as the class priors were $P(L = 0) = 0.7$ and $P(L = 1) = 0.3$. This is the sample data I use to answer the questions in this section and is shown in Figure 1.

1.A Expected Risk Minimization (ERM) Classification

Classification using the knowledge of true data PDF.

1.A.1 Minimum expected risk classification rule

The following is the minimum expected risk classification rule in the form of a likelihood-ratio test:

$$\frac{p(x|L=1)}{p(x|L=0)} = \frac{g(x|m_1, C_1)}{p(x|m_0, C_0)} > \gamma = \frac{p(x|L=0)}{p(x|L=1)} * \frac{\lambda_{01} - \lambda_{00}}{\lambda_{10} - \lambda_{11}}$$

where the threshold γ is a function of class priors and fixed (nonnegative) 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. λ_{ij} is defined in Table 1.

λ_{ij} values range from 0 to 1, where 1 is the highest cost. In order to minimize expected risk, the costs

Conditional Loss Function	
λ_{ij}, where $\lambda_{ij} = \lambda(\alpha_i \vert \omega_j)$	Cost Type
λ_{00}	True negative
λ_{01}	False negative
λ_{10}	False positive
λ_{11}	True positive

Table 1: The type of cost given by the conditional loss functions, λ_{ij}, in the likelihood-ratio test.

for incorrect results (false negatives and positives), λ_{01} and λ_{10}, are set to the highest cost possible, 1, while the costs for correct results (true negatives and positives), λ_{00} and λ_{11}, are set to the lowest cost possible, 0.

$$\therefore \gamma = \frac{0.7}{0.3} * \frac{1-0}{1-0} = 2.333$$

$$\therefore \frac{p(x|L=1)}{p(x|L=0)} > 2.333$$

1.A.2 Classifier Implementation and the ROC Curve

Figure 2 shows the ROC curve for the implemented ERM based classification.

1.A.3 Experimental and Theoretical Minimum Risk

I first calculated the theoretical false positive, $P(D=1|L=0;\gamma)$, and true positive, $P(D=1|L=1;\gamma)$. I then used the true positive to calculate true negative, $1 - P(D=1|L=1;\gamma) = P(D=0|L=1;\gamma)$. To calculate the theoretical minimum risk and I used the following formula:

$$P(error;\gamma) = P(D=1|L=0;\gamma) * P(L=0) + P(D=0|L=1;\gamma) * P(L=1).$$

Next, I retrieved the experimental values that I held when implementing my classifier previously. Finally, I

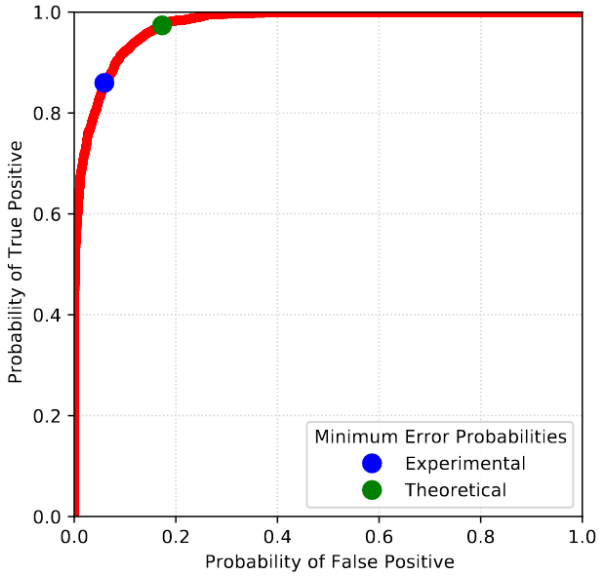

Figure 2: *The ROC curve with the minimum theoretical and experimental error probabilities.*

Comparing Optimal Thresholds		
Type of Estimate	Gamma, γ	P(error)
Theoretical	2.333	0.082
Experimental	2.266	0.084

Table 2: Comparison of gammas and probability of errors that are the values for the minimum theoretical and experimental errors in the classifier implementation.

plotted the theoretical and experimental values on the ROC curve, shown in Figure 2. The experimental and theoretical minimum errors are slightly different, ≈ 0.04, but similar enough to conclude they are accurate. In Table 2, the gammas for the minimum error probabilities are compared.

1.B Naive Bayesian Classifier

For this part, I implemented the ERM classification using incorrect knowledge of data distribution (Naive Bayesian Classifier). I repeated the sampling from Part A but replaced the covariance matrices with ones that are diagonal with diagonal entries equal to true variances, off-diagonal entries equal to zeros, as such:

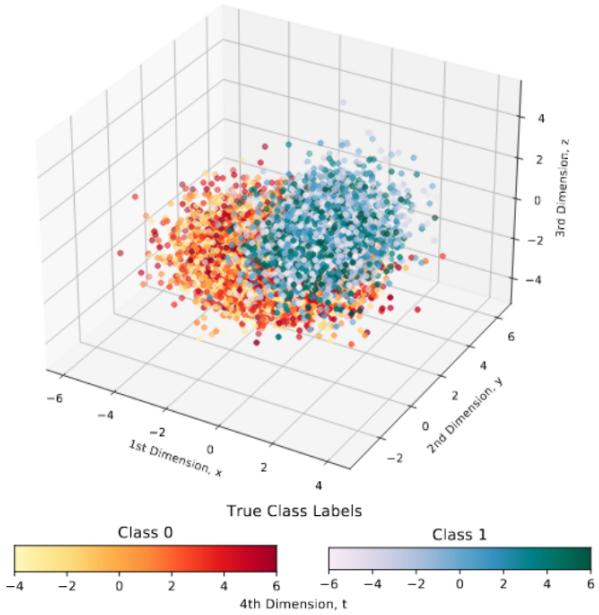

Figure 3: *10,000 sample values from the provided multivariate Gaussian probability density function distribution with two true classes and incorrect, independent covariance matrices. As higher dimensions are difficult to visualize, I use a gradient colorbar to indicate the value of the fourth dimension.*

$$\sigma_0^2 = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \sigma_1^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

The samples from the modified multivariate Gaussian PDF based on Naive Bayesian classification are visualized in Figure 3. Visually, the samples form "looser" clusters than the previous non-Naive samples.

1.B.1 Minimum expected risk classification rule

This remains the same as Part A, as the value of the variables are unchanged.

$$\therefore \frac{p(x|L=1)}{p(x|L=0)} > \gamma = 2.333$$

1.B.2 Classifier Implementation and the ROC Curve

Figure 4 shows the ROC curve for the implemented ERM based classification using a Naive Bayesian classifier.

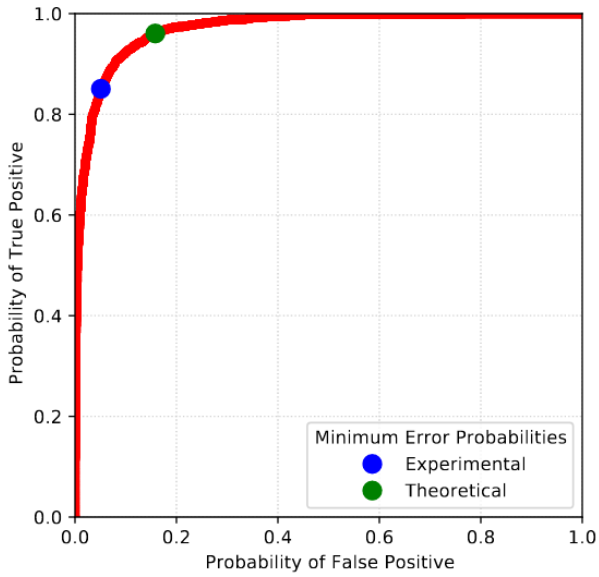

Figure 4: *The ROC curve with the minimum theoretical and experimental error probabilities in the Naive Bayesian classifier implementation.*

Comparing Optimal Thresholds		
Type of Estimate	Gamma, γ	P(error)
Theoretical	2.333	0.082
Experimental	2.429	0.081

Table 3: Comparison of gammas and probability of errors that are the values for the minimum theoretical and experimental errors in the Naive Bayesian classifier implementation.

1.B.3 Experimental and Theoretical Minimum Risk

The methodology for calculating the gammas and minimum error probabilities is the same as Part A. In comparing these values to the non-Naive classifier (Table 3 and Table 2), only an insignificant difference occurs. This model mismatch did not negatively impact the ROC curve, and slightly decreased both the theoretical and experimental probability of errors.

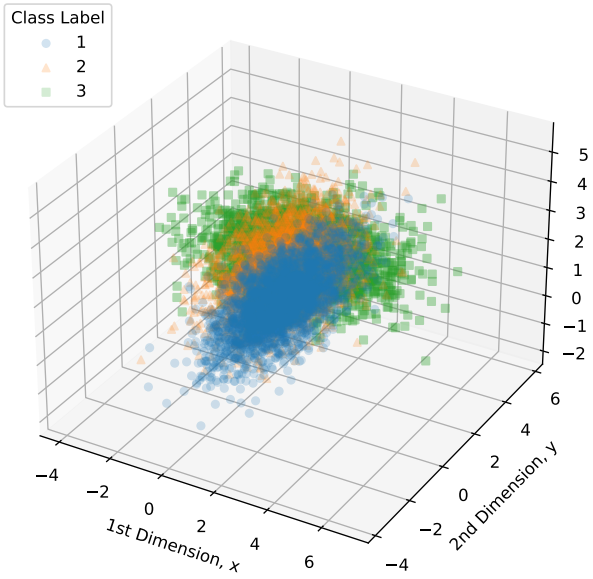

Figure 5: *10,000 sample values from the provided multivariate Gaussian probability density function distribution with three true classes and their specified mean vectors and covariance matrices.*

2

For this section, I create a distribution with 3 classes created from 4 Gaussians. The following are the class priors and parameters for the class-conditional pdfs:

$$p(x|L=1) : p(L=1) = 0.3, \mu = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \sigma^2 = \begin{bmatrix} 0.5 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 0.5 & 0.5 \end{bmatrix}$$

$$p(x|L=2) : p(L=2) = 0.3, \mu = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}, \sigma^2 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0.5 & 1 \end{bmatrix}$$

$$p(x|L=3)^1 : p(L=1) = 0.4, \mu = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}_a \; or \; \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}_b, \sigma^2 = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}_a \; or \; \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}_b$$

2.A Bayes Decision Rules, "MAP Classifier"

Gives the minimum probability of error classification with 0-1 loss.

[1] Class 3 data originates from a mixture of the remaining 2 Gaussian components, labeled a and b, with equal weights.

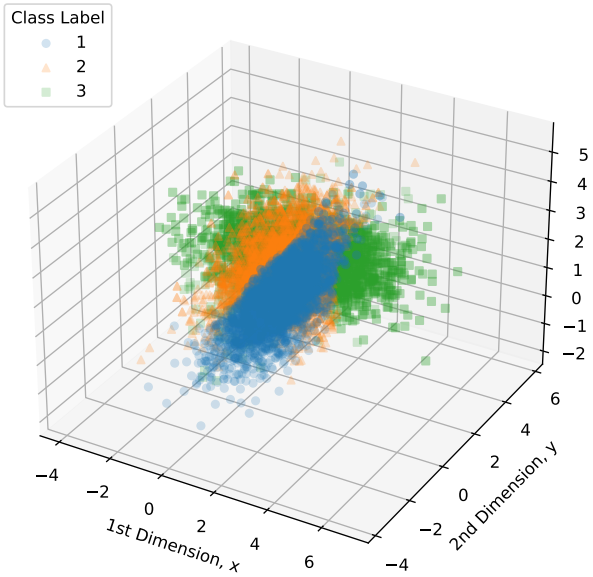

Figure 6: *The classes in which the 10,000 samples were classified using the given decision rule with a 0-1 loss.*

2.A.1 Sampling from the Distribution

Figure 5 is the 10,000 samples taken from the distribution defined at the beginning of this section.

2.A.2 Decision Rule and Confusion Matrix

A decision rule that achieves the minimum probability of error is the 0-1 loss, which is a special case decision rule as follows[2]:

$$R(\alpha_i|x) = \sum_{j=1}^{C} \lambda(\alpha_i|\omega_j) * P(\omega_j|x) = \sum_{j \neq i} P(\omega_j|x)$$

Figure 6 shows how the samples were classified according to this decision rule. Additionally, I estimated the normalized confusion matrix, which is shown in Figure 7.

2.A.3 Accuracy of Classification

Figure 8 provides a visualization of whether the data was correctly or incorrectly classified.

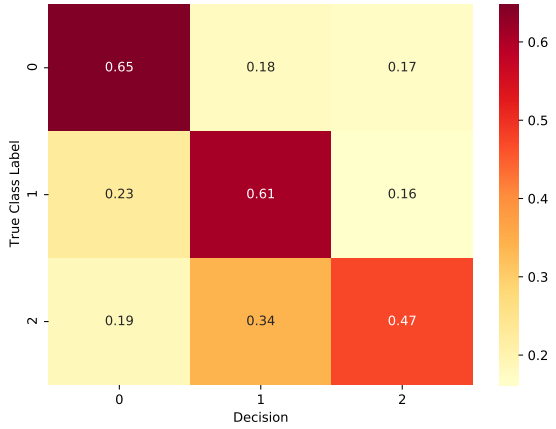

Figure 7: *Decision matrix normalized over the true values for the classification done by the decision rule using a 0-1 loss.*

2.B Modified Loss Matrices

2.B.1 Decision Rules and Confusion Matrices

Since we longer have special cases for the decision rule, the following is the decision rule I used:

$$R(\alpha_i|x) = \sum_{j=1}^{C} \lambda(\alpha_i|\omega_j) * P(\omega_j|x)$$

Figures 9 and 10 show how the samples were classified according to this decision rule when the given decision rule cares 10 or 100 times more about not making mistakes when L=3, respectively. Additionally, Figures 11 and 12 show the normalized confusion matrices.

2.B.2 Accuracy of Classification

Figures 13 and 14 provide a visualization of whether the data was correctly or incorrectly classified when the given decision rule cares 10 or 100 times more about not making mistakes when L=3, respectively.

2.B.3 Interesting Insights

When modifying the loss matrix and weighing the Class 3 with more risk, the classifier had more errors and classified significantly more data as Class 3 (whether it was correct or not). For example, when the decision rule cares 10 times more about not making a mistake when L=3, we see the majority of data being classified in Class 3 (as shown in Figure 9). Shifting accuracy for the classification of the individual classes can be observed in the confusion matrices (Figures 7, 11, and 12). The accuracy for classifying Class 3 does not improve when the classifier higher how much it cares about not making a mistake when L=3 from 10 times to 100 times as the accuracy for Class 3 at 10 times is already 100%. Additionally, we see that as the weight of risk for Class 3, the classifier shifts the decisions for Class 1 and 2 more towards the right-most column, which is the Class decision.

[2]C being the length of the set of classes

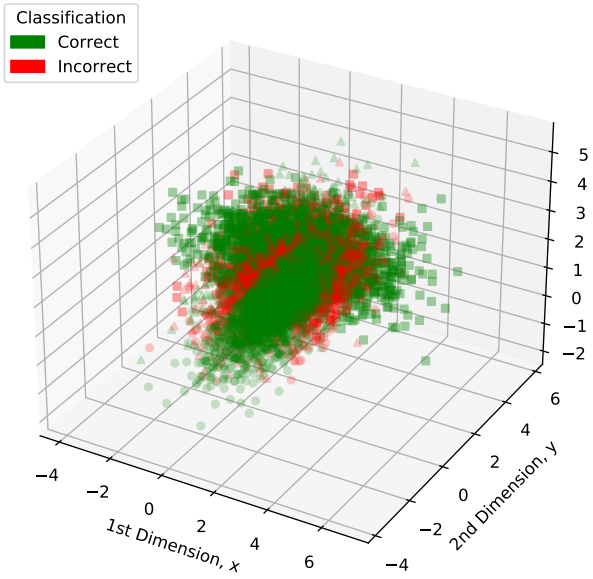

Figure 8: *Whether the samples were classified correctly or incorrectly using the given decision rule with a 0-1 loss.*

This behavior variation, when the loss matrix is modified, is expected and only increases when the decision rule is set to care 100 times more about not making a mistake when L=3. In fact, when Class 3 has such high risk, almost no data was classified in Class 1 or 2 (as shown in Figure 10). This might be beneficial in some instances, but results in a significant amount of incorrect classification in Class 1 and 2 (as shown in Figures 10 and 11).

3 ERM Classifier Applications

3.A Wine Quality

In this section, I implement a minimum-probability-of-error classifier to classify the quality of wines based on 11 physical attributes. Figures 15 and 16 show two visualizations for two subsets of the data citric acid, total sulfer dioxide, and density and alcohol, pH, and residual sugar. My first step was to calculate an estimation of the covariance matrices, mean vectors, and class priors. I did this by separating the data by true class label (wine quality, in this case). To calculate mean vector, I simply found the means of each of the 11 attributes. To find the class prior, I divided the number of samples in a class by the total number of samples. To calculate the covariances, I used the following formula which uses 1 degree of freedom:

$$cov_{x,y} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

Next, I implemented my classifier. When classifying each sample, I chose the class label with the minimum risk. As done in the previous section, I used the following risk algorithm for my decision rule:

$$R(\alpha_i|x) = \sum_{j=1}^{C} \lambda(\alpha_i|\omega_j) * P(\omega_j|x)$$

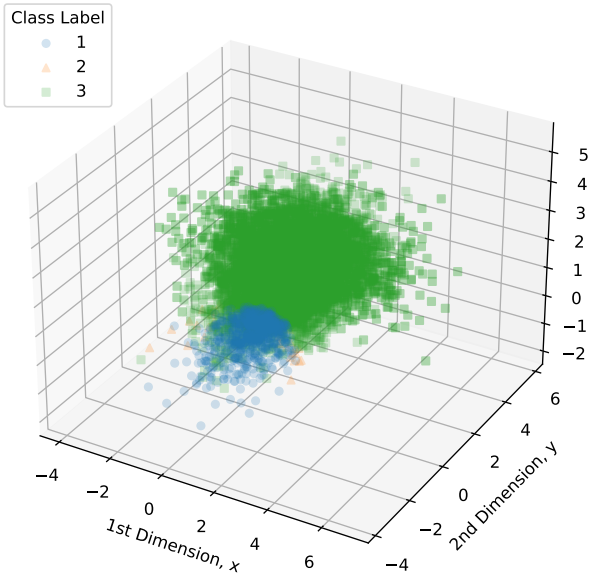

Figure 9: *The classes in which the 10,000 samples were classified using the given decision rule the cares 10x more about not making mistakes when L=3.*

In Figures 17 and 18, we are shown the data that is correctly and incorrectly classified. The classifier has an accuracy of 45%. In Figure 19, the confusion matrix, we see that the classifier classified all the data into one class. I did attempt to modify the loss matrix to care significantly more if the outliers were incorrectly classified. However, it did not improve the performance of my classifier. The following is the loss matrix I tried to use:

$$\lambda = \begin{bmatrix} 0 & 15 & 20 & 25 & 30 & 35 & 40 & 45 & 50 \\ 15 & 0 & 10 & 15 & 20 & 25 & 30 & 35 & 40 \\ 20 & 10 & 0 & 5 & 10 & 15 & 20 & 25 & 30 \\ 25 & 15 & 5 & 0 & 1 & 5 & 10 & 15 & 20 \\ 30 & 20 & 10 & 1 & 0 & 1 & 1 & 5 & 10 \\ 35 & 25 & 15 & 5 & 1 & 0 & 10 & 15 & 20 \\ 40 & 30 & 20 & 10 & 1 & 10 & 0 & 25 & 30 \\ 45 & 35 & 25 & 15 & 5 & 15 & 25 & 0 & 40 \\ 50 & 40 & 30 & 20 & 10 & 20 & 30 & 40 & 0 \end{bmatrix},$$

Due to the extremely poor and biased performance of the classifier with this dataset, I would not recommend using it for such.

3.B Human Activity

In this section, I implement a minimum-probability-of-error classifier to classify the quality of human activity recognition based on 561 attributes using a 0-1 loss. Figures 20 and 21 show two visualizations for two subsets of the data aa, ab, and ac and Yh, Yi, and Yj. My first step was to calculate an estimation of the covariance matrices, mean vectors, and class priors. I did this by separating the data by true class label (wine quality,

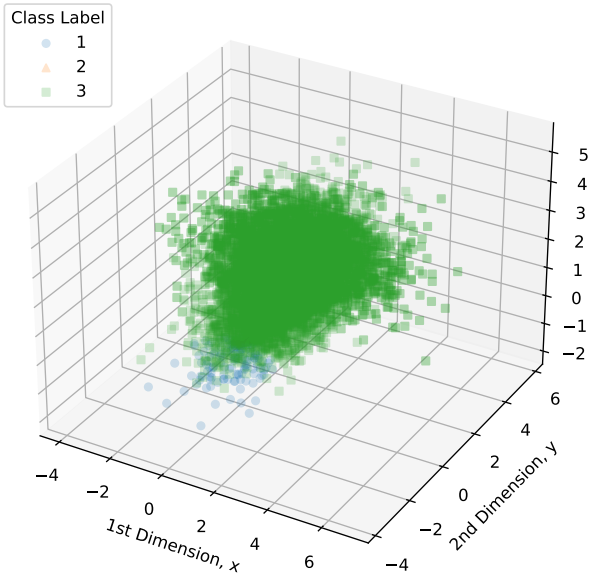

Figure 10: *The classes in which the 10,000 samples were classified using the given decision rule cares 100x more about not making mistakes when L=3.*

in this case). To calculate mean vector, I simply found the means of each of the 561 attributes. To find the class prior, I divided the number of samples in a class by the total number of samples. To calculate the covariances, I used the following formula which uses 1 degree of freedom:

$$cov_{x,y} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

Next, I implemented my classifier. When classifying each sample, I chose the class label with the minimum risk. As done in the previous section, I used the following risk algorithm for my decision rule:

$$R(\alpha_i|x) = \sum_{j=1}^{C} \lambda(\alpha_i|\omega_j) * P(\omega_j|x)$$

In Figures 22 and 23, we are shown the data that is correctly and incorrectly classified. The classifier has an accuracy of 68%. In Figure 24, the confusion matrix, we see that the classifier classified the data exceptionally well for some labels and decently for others. I believe with tweaking the loss matrix, the performance may improve enough for this model to be usable with this data, as I stuck to using 0-1 loss for this dataset.

4 Appendix

See my GitHub for all the source code: Click Me!
or copy and paste:

`https://github.com/emilyjcosta5/machine_learning/tree/main/homework_1`

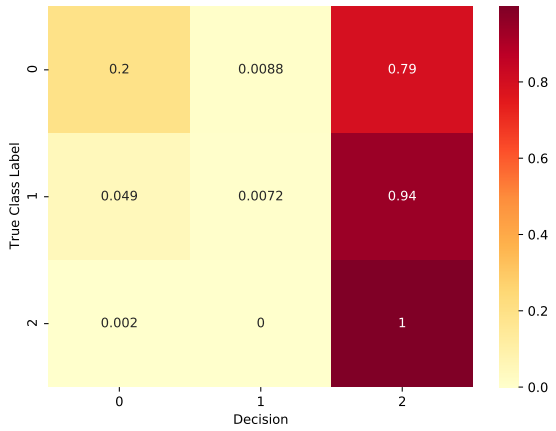

Figure 11: *Decision matrix normalized over the true values for the classification done by the decision rule that cares 10 times more about not making mistakes when L=3.*

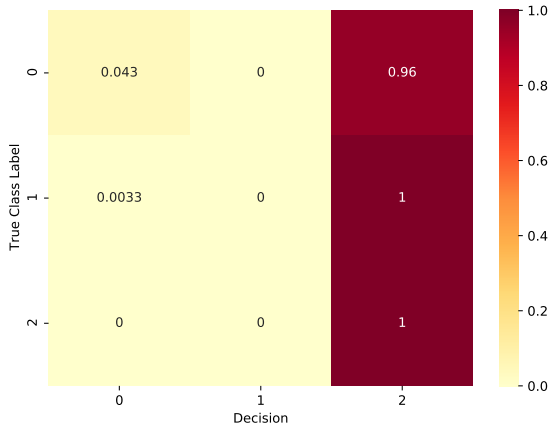

Figure 12: *Decision matrix normalized over the true values for the classification done by the decision rule that cares 100 times more about not making mistakes when L=3.*

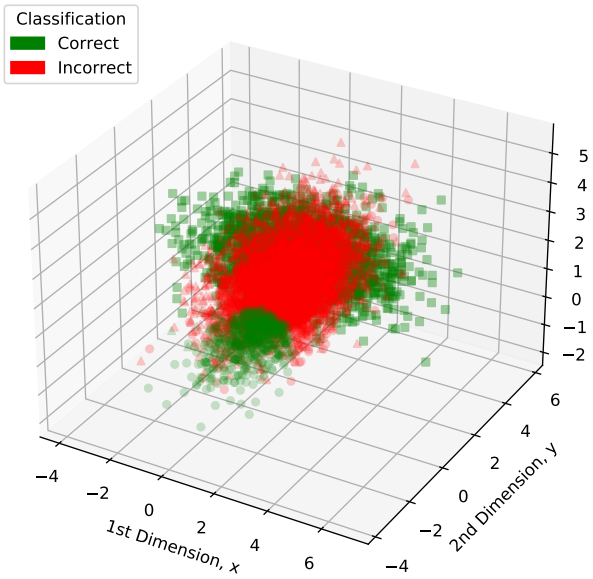

Figure 13: *Whether the samples were classified correctly or incorrectly using the given decision rule that cares 10x more about not making mistakes when L=3.*

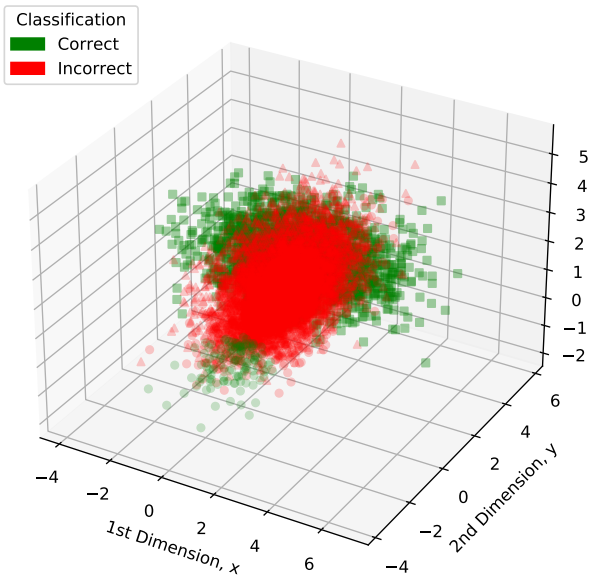

Figure 14: *Whether the samples were classified correctly or incorrectly using the given decision rule that cares 100x more about not making mistakes when L=3.*

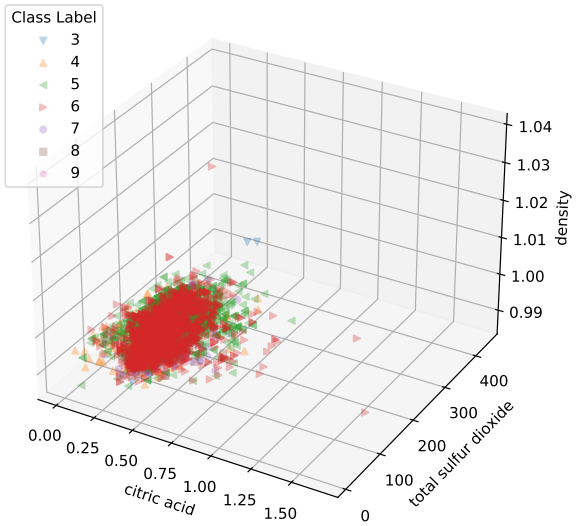

Figure 15: *A subset- citric acid, total sulfer dioxide, and density- of the wine attributes shown by true class label.*

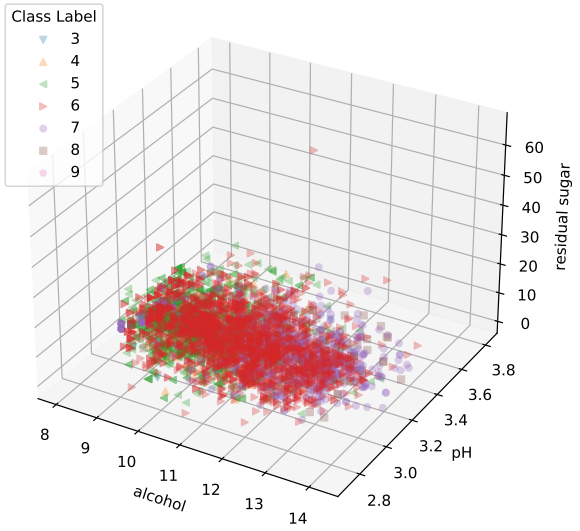

Figure 16: *A subset- alcohol, pH, and residual sugar- of the wine attributes shown by true class label.*

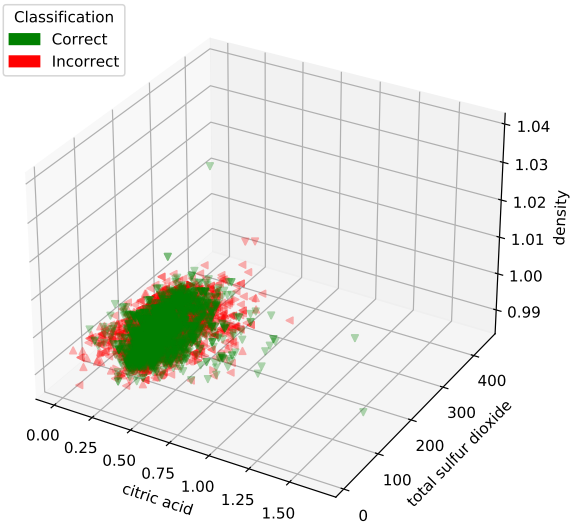

Figure 17: *A subset- citric acid, total sulfer dioxide, and density- of the wine attributes shown by true class label and colored by whether the data was classified correctly or not.*

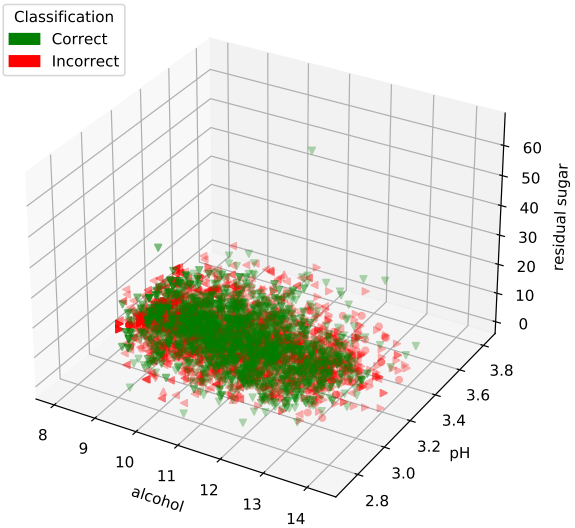

Figure 18: *A subset- alcohol, pH, and residual sugar- of the wine attributes shown by true class label and colored by whether the data was classified correctly or not.*

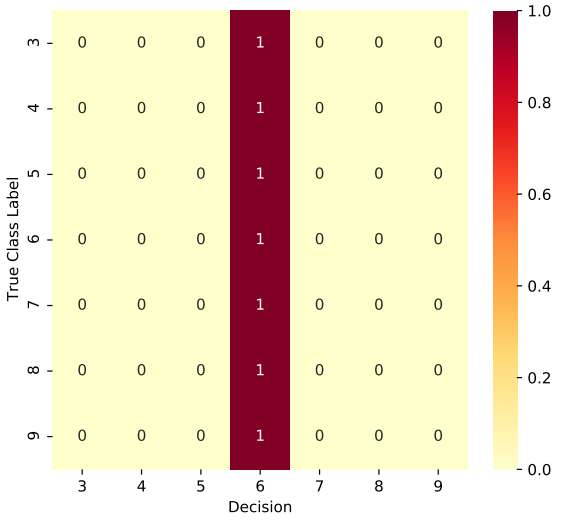

Figure 19: *The confusion matrix for the wine dataset*

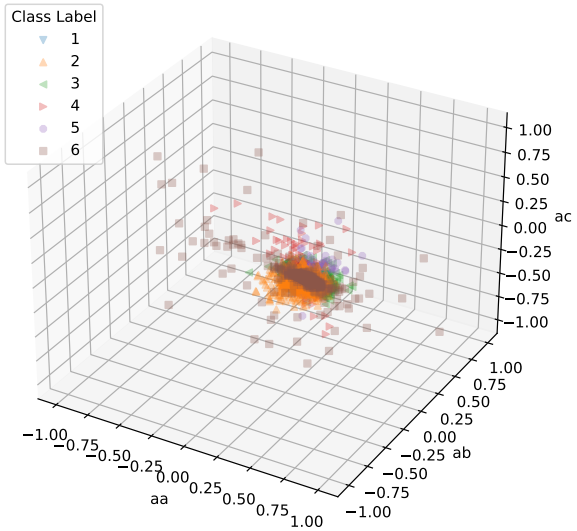

Figure 20: *A subset- citric acid, aa, ab, and ac (labels which I made up)- of the human activity recognition attributes shown by true class label.*

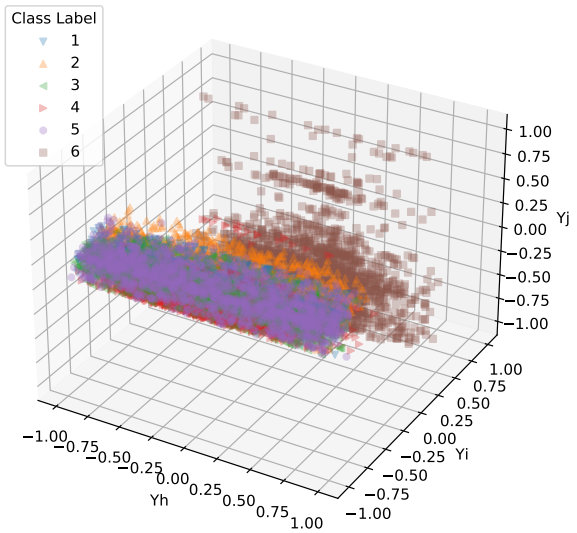

Figure 21: *A subset- citric acid, Yh, Yi, and Yj (labels which I made up)- of the human activity recognition attributes shown by true class label.*

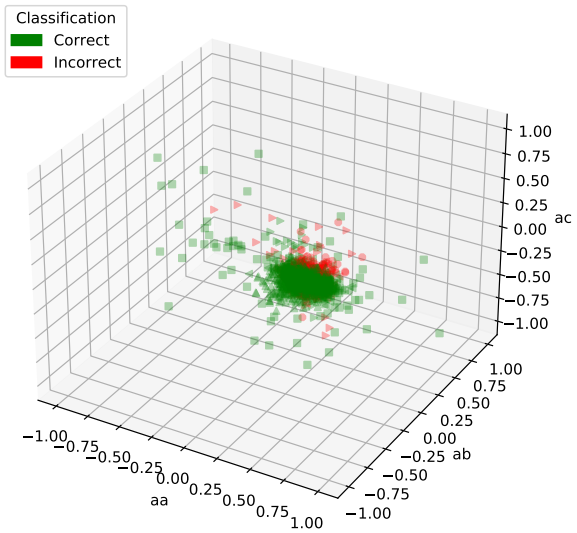

Figure 22: *A subset- citric acid, aa, ab, and ac (labels which I made up)- of the human activity recognition attributes shown by true class label and colored by whether the data was classified correctly or not.*

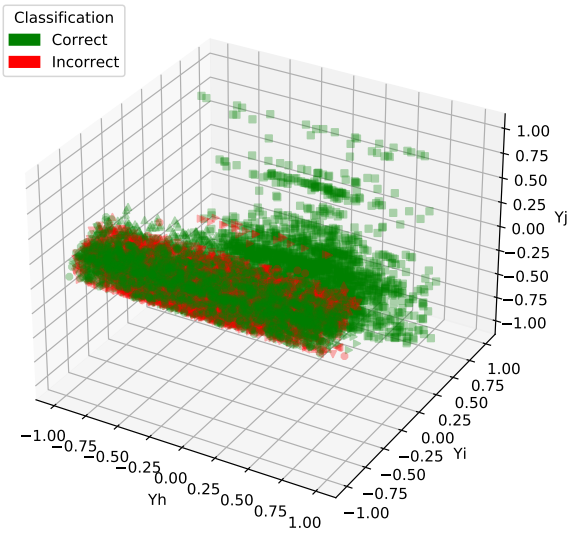

Figure 23: *A subset- citric acid, Yh, Yi, and Yj (labels which I made up)- of the human activity recognition attributes shown by true class label and colored by whether the data was classified correctly or not.*

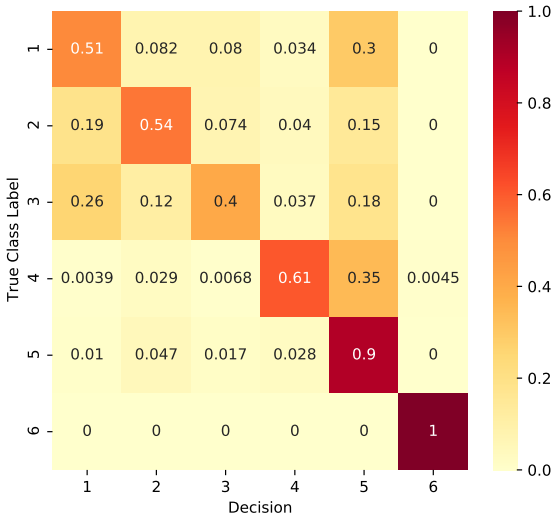

Figure 24: *The confusion matrix for the human activity recognition dataset*