Question 1 (30%)

The probability density function (pdf) for a 3-dimensional real-valued random vector $\mathbf{X}$ is as follows: $p(\mathbf{x}) = p(\mathbf{x}|L=0)p(L=0) + p(\mathbf{x}|L=1)p(L=1)$. Here L is the true class label that indicates which class-label-conditioned pdf generates the data.

The class priors are $p(L=0) = 0.65$ and $p(L=1) = 0.35$. The class-conditional pdfs are $p(\mathbf{x}|L=0) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ and $p(\mathbf{x}|L=1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, where $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a multivariate Gaussian probability density function with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The parameters of the class-conditional Gaussian pdfs are:

$$\boldsymbol{\mu}_0 = \begin{bmatrix} -1/2 \\ -1/2 \\ -1/2 \end{bmatrix} \quad \boldsymbol{\Sigma}_0 = \begin{bmatrix} 1 & -0.5 & 0.3 \\ -0.5 & 1 & -0.5 \\ 0.3 & -0.5 & 1 \end{bmatrix} \quad \boldsymbol{\mu}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0.3 & -0.2 \\ 0.3 & 1 & 0.3 \\ -0.2 & 0.3 & 1 \end{bmatrix}$$

To produce numerical results as requested below, generate $N = 10000$ samples according to this data distribution. Keep track of the true class labels for each sample. Use the same dataset in all the 'Parts' described below.

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

1. Specify the minimum expected risk classification rule in the form of a likelihood-ratio test: $\frac{p(\mathbf{x}|L=1)}{p(\mathbf{x}|L=0)} \overset{?}{>} \gamma$, where the threshold γ is a function of class priors and fixed (non-negative) loss values. These loss values (written as λ_{ij} in lecture notes) apply to each of the four cases $D = i\,|\,L = j$, where $i, j \in 0, 1$ and D is the decision label.

2. Implement this classifier and apply it to the 10K samples you generated. Vary the threshold γ gradually from 0 to ∞, and for each threshold value 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 $\gamma \approx 0$ the ROC curve should be at $\binom{1}{1}$, and as γ increases it should traverse towards $\binom{0}{0}$. 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, as you will use these values in the following parts.

3. Determine and report the threshold value that achieves minimum probability of error. Superimpose clearly on the ROC curve (using a different color/shape marker) the true-positive and false-positive values attained by this minimum-Pr(error) classifier. Calculate and report an estimate of the minimum probability of error that is achievable for this data distribution. Note that $\Pr(\text{error};\gamma) = p(D=1|L=0;\gamma)p(L=0) + p(D=0|L=1;\gamma)p(L=1)$, where you should use sample estimates of class priors $p(L=j)$, like in the Notebook examples, as you wish to evaluate error rates based on the dataset generated. How does your empirically selected γ value that minimizes Pr(error) compare with the theoretically optimal threshold you compute from the priors and loss values?

Part B: ERM classification attempt using incorrect knowledge of data distribution.

For this part, you will implement a Naive Bayes Classifier, which assumes features are independent given each class label. Let's 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 (correct) true means described above, but (incorrect) covariance matrices. In particular, you choose for both covariance matrices to be equal to the identity matrix (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 using the Naive Bayes classifier design by repeating the same steps in **Part A** on the same 10K sample dataset 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?

Part C: Fisher's Linear Discriminant Analysis classification.

In the third part of this exercise, repeat the same steps as in the previous two parts, but this time using a Fisher's Linear Discriminant Analysis (LDA) classifier. Given the available 10K samples, estimate the class-conditional pdf mean and covariance matrices using sample average estimators (*i.e.* the estimated mean vector and covariance matrix from the 10K samples). From these sample-based estimates of mean and covariance, determine the Fisher's LDA projection weight vector (via the generalized eigendecomposition of within- and between-class scatter matrices): $\mathbf{w}_{LDA}$. Trace the ROC curve for the classification rule $\mathbf{w}_{LDA}^T \mathbf{x}$ compared against a threshold γ that varies from $-\infty$ to ∞. 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. Discuss how this LDA classifier performs relative to the previous two classifiers.

Note: When finding the Fisher's LDA projection matrix, do not be concerned about the difference in class priors. When determining the between-class and within-class scatter matrices, use equal weights for the class means and covariances, like we did in class.

Question 2 (30%)

A 2-dimensional random vector $\mathbf{X}$ takes values from a mixture of four Gaussians. Each Gaussian pdf is the class-conditional pdf for one of four class labels $L \in \{1,2,3,4\}$. For this problem, pick your own 4 distinct Gaussian class-conditional pdfs $p(\mathbf{x}|L=j),\ j \in \{1,2,3,4\}$. Set class priors to $0.2, 0.25, 0.25, 0.3$. Select your Gaussian class-conditional pdfs to have mean vectors approximately equally spaced out along a line, and the covariance matrices to be scaled versions of the identity matrix (with scale factors leading to a significant amount of overlap between the data from these Gaussians). Label these classes according to the ordering of mean vectors along the line, so that classes have consecutive integer labels.

Part A: Minimum Probability of Error classification (0-1 loss, also referred to as Bayes Decision rule or Maximum a Posteriori classifier).

1. Generate $N = 10000$ samples from this data distribution and keep track of the true labels for each sample.
2. Specify the decision rule that achieves minimum probability of error (*i.e.* use 0-1 loss). Implement this classifier with true data distribution knowledge. Then classify the 10K samples and count the samples corresponding to each decision-label pair so as to empirically estimate the confusion matrix whose entries are $p(D=i|L=j)$ for $i,j \in \{1,2,3,4\}$. Present results of the confusion matrix and minimum probability of error estimate.
3. Provide a visualization of the data (scatter plot in 2-dimensional space). 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 using the marker colors indicated in parentheses.

Part B: ERM classification (loss matrix values that do not correspond to 0-1 loss).

Repeat the exercise for an ERM classification rule with the following loss values:

$$\Lambda = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{bmatrix} \tag{1}$$

Given this loss matrix, errors between Gaussian pairs that have higher separation in their means will be penalized more.

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. Again present your results with visual and numerical representations. Briefly discuss interesting insights, if any.

Hint: For each sample, determine the loss matrix entry corresponding to the decision-label pair that this sample falls into, and add this loss to an estimate of cumulative loss. Divide cumulative loss by the total number of samples to get average loss as an estimate for expected risk/loss.

Question 3 (40%)

Download the following datasets:

- Wine Quality dataset located at `https://archive.ics.uci.edu/ml/datasets/Wine+Quality`, specifically the **white** wine dataset. This dataset consists of 11 features and class labels from 0 to 10 indicating wine quality scores. There are 4898 samples.
- Human Activity Recognition dataset found at `https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones`. This dataset 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 is Gaussian. Using all available samples from a class, compute sample average estimates for the mean vectors and covariance matrices that parameterize the class-conditional Gaussians. Use sample counts to also estimate the class priors. In case your sample estimates of covariance matrices are ill-conditioned[1], consider adding a regularization term to your covariance estimate, *e.g.* $\mathbf{\Sigma}_{Regularized} = \mathbf{\Sigma}_{SampleAverage} + \lambda\,\mathbf{I}$, where $\lambda > 0$ is a small regularization parameter that ensures the regularized covariance matrix $\mathbf{\Sigma}_{Regularized}$ has all eigenvalues larger than this parameter. Using regularization in this context will allow you to solve an otherwise ill-posed problem.

With these estimated ("trained") Gaussian class-conditional pdfs and class priors, apply the minimum-Pr(error) classification rule on all ("training") samples. Then count the errors and report the error probability estimate you obtain for each problem, as well as the confusion matrices for this classification rule.

Visualize the datasets in 2- or 3-dimensional projections of subsets of features and then do the same 2- or 3-dimensional plots using principal component analysis (PCA). Discuss the following:

- If Gaussian class-conditional models are appropriate for these datasets, commenting on the differences in how feature-subsets or PCA helped you draw your conclusions
- How your model choice of a Gaussian might have influenced the confusion matrix and probability of error values you obtained
- Any modeling assumptions, *e.g.* how you estimated/selected necessary parameters for your model and classification rule

Describe your analyses in mathematical terms supplemented by numerical and visual results, conveying your understanding of what you have accomplished and demonstrated.

Hint: Later in the course, we will talk about how to select regularization parameters. For now, you may consider using a value on the order of an arithmetic average of the sample covariance matrix estimate's non-zero eigenvalues: $\lambda = \alpha \times trace(\mathbf{\Sigma}_{SampleAverage})/rank(\mathbf{\Sigma}_{SampleAverage})$ *or a geometric average of the sample covariance matrix estimate's non-zero eigenvalues, where* $0 < \alpha < 1$ *is a small real number. This makes your regularization term proportional to the eigenvalues observed in the sample covariance estimate.*

[1] Read here about ill-conditioned matrices: `https://deepai.org/machine-learning-glossary-and-terms/ill-conditioned-matrix`