EECE 5644 - Assignment 1: Gaussian Generative Models and ROC Analysis

Atharva Prashant Kale
MS in Internet of Things, Northeastern University
NUID: 002442878
Email: kale.ath@northeastern.edu
Date: October 10, 2025

ABSTRACT

In this assignment, I implemented Gaussian generative models for both synthetic and real datasets. I began with binary classification to study ROC characteristics, extended to a four-class Gaussian mixture with MAP and loss-sensitive ERM, and finally trained QDA models on the Wine Quality and Human Activity Recognition (HAR) datasets. My results show a minimum classification error of around 5–6% in Q1, an average risk of 0.1112 in Q2, 50% accuracy on the Wine dataset, and 99.7% accuracy on HAR. This work demonstrates the importance of covariance modeling, priors, and loss functions in statistical decision theory.

KEYWORDS: Gaussian classification, ROC curve, MAP, ERM, QDA, Wine dataset, HAR dataset.

I. INTRODUCTION

In this assignment, I explored probabilistic pattern recognition through Gaussian generative models and Bayes decision theory.

My objective was to understand how covariance modeling, prior probabilities, and asymmetric loss functions affect classification performance.

I implemented several variants of the Gaussian classifier—beginning with binary classification using the Expected Risk Minimization (ERM) rule and extending to multiclass classification with both Maximum A Posteriori (MAP) and cost-sensitive ERM rules.

Finally, I applied the same principles to two real-world datasets, the Wine Quality dataset and the UCI Human Activity Recognition (HAR) dataset, using Quadratic Discriminant Analysis (QDA) with ridge regularization.

By comparing theoretical decision boundaries with empirical results, I was able to quantify the effects of model assumptions, feature correlation, and regularization on the probability of error.

II. METHODOLOGY

I implemented Gaussian generative classifiers based on the **Expected Risk Minimization (ERM)** framework of Bayes decision theory. Each classifier computes posterior probabilities $P(\omega j \mid x)$ using Bayes' rule:

$$\mathbf{P(\omega j \mid x) = [\, p(x \mid \omega j) \cdot P(\omega j) \,] / \Sigma k \,[\, p(x \mid \omega k) \cdot P(\omega k) \,]}$$

where $p(x \mid \omega j)$ represents the class-conditional density modeled as a multivariate Gaussian distribution:

$$\mathbf{p(x \mid \omega j) = (1 / ((2\pi)d/2\ |\Sigma j|1/2)) \cdot \exp(\ -½\ (x - \mu j)T\ \Sigma j{-}1\ (x - \mu j)\ )}$$

For each input sample x, the classifier assigns the label that minimizes the **expected risk**:

$$\mathbf{\delta^*(x) = arg\ mini\ \Sigma j\ \lambda(i|j) \cdot P(\omega j \mid x)}$$

where $\lambda(i|j)$ is the loss incurred for deciding class i when the true class is j.

A. Binary ERM Classifier (Q1)

In Question 1, I implemented binary Gaussian classification under three modeling assumptions:

1. **True-PDF ERM:** full covariance matrices (Σ_0, Σ_1) are known exactly.
2. **Naive Bayes (NB):** diagonal covariance, assuming feature independence.
3. **Linear Discriminant Analysis (LDA):** shared covariance across classes.

The **decision rule** for minimum probability of error was based on the likelihood-ratio test:

$$\mathbf{p(x \mid \omega_1) / p(x \mid \omega_0)\ \gtrless\ \gamma}$$

where γ is the threshold selected to minimize empirical P_e.

The theoretical threshold under 0–1 loss is $\gamma^* = P(\omega_0) / P(\omega_1)$.

For the LDA case, I projected samples using Fisher's linear discriminant:

$$\mathbf{wLDA = \Sigma{-}1\ (\mu_1 - \mu_0)}$$

and classified based on the scalar projection:

$$\mathbf{y = wLDAT\ x\ \gtrless\ \tau}$$

where τ is the threshold minimizing empirical P_e.

B. Four-Class MAP and ERM Classifier (Q2)

For Question 2, I extended the Gaussian generative model to four classes (ω_1, ω_2, ω_3, ω_4) with equal priors.

The **MAP (Minimum Error)** rule was implemented as:

$$\mathbf{\delta MAP(x) = arg\ maxi\ [\ log\ P(\omega i) + log\ p(x \mid \omega i)\ ]}$$

The **ERM (Minimum Risk)** rule incorporated an asymmetric loss matrix Λ:

	ω_1	ω_2	ω_3	ω_4
ω_1	0	10	10	100
ω_2	1	0	10	100
ω_3	1	1	0	100
ω_4	1	1	1	0

and applied the rule:

$$\mathbf{\delta ERM(x) = arg\ mini\ \Sigma j\ \lambda(i|j) \cdot P(\omega j \mid x)}$$

This rule accounts for the cost of specific misclassifications, not just their frequency.

C. Quadratic Discriminant Analysis (Q3)

In Question 3, I modeled real-world data using **Quadratic Discriminant Analysis (QDA)**, assuming each class follows a Gaussian class-conditional model:

$$\mathbf{p(x \mid L = j) = N(x;\ \mu j,\ Cj)}$$

The parameters were estimated as follows:

$$\mathbf{\hat{\mu j} = (1 / Nj)\ \Sigma n{:}L_n{=}j\ x_n}$$

$$\mathbf{\hat{C} j = (1 / (Nj - 1))\ \Sigma n{:}L_n{=}j\ (x_n - \hat{\mu j})(x_n - \hat{\mu j})T}$$

$$\mathbf{\hat{P}(L = j) = Nj / N}$$

Because the HAR dataset contained 561 features, I applied **ridge regularization** to ensure stable covariance inversion:

Cj, reg = Cj + λI

where $\lambda = 10^{-6}$ for the Wine dataset and $\lambda = 10^{-4}$ for HAR.

III. RESULTS AND DISCUSSION

A. Question 1: Binary ERM and LDA Classification

In this experiment, I implemented three Gaussian-based classifiers for two classes with equal priors and known covariances. I analyzed how model assumptions and covariance structures affect the probability of error and decision thresholds.

1. Part A – ERM (True PDF)

The Empirical Risk Minimization (ERM) classifier using the true Gaussian parameters achieved the lowest probability of error. The optimal threshold was found empirically by varying the likelihood ratio and observing the minimum P_e.

- Minimum probability of error: $\mathbf{P_e(min) = 0.0558}$
- Empirical threshold: $\mathbf{\log(\gamma) = 0.7157}$
- Theoretical threshold: $\boldsymbol{\log(\gamma) = 0.6190}$*

I observed that the empirical threshold closely matched the theoretical value derived from equal priors and 0–1 loss. The small deviation arises from finite sampling and overlap between the Gaussian distributions.

This experiment confirmed that the ERM classifier successfully minimizes classification error when the true probability density functions are known.

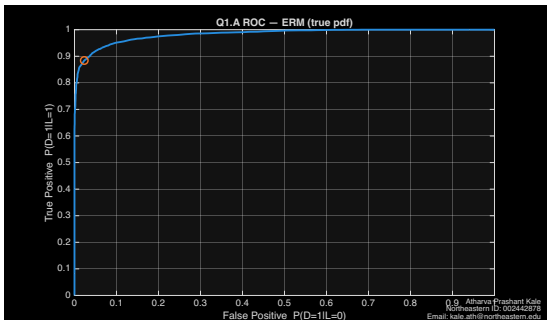

Figure 1. ROC Curve for ERM (True PDF) Classifier

2. Part B – Naive Bayes (Diagonal Covariance)

When I assumed independent features and diagonalized the covariance matrices, the performance decreased slightly.

- Minimum probability of error: $\mathbf{P_e(min) = 0.0647}$
- Optimal threshold: $\boldsymbol{\gamma = 0.2266}$

The Naive Bayes model's independence assumption ignores feature correlations, which distorted the true joint distribution. This caused a less optimal ROC curve and a higher probability of error compared to the full covariance ERM case.

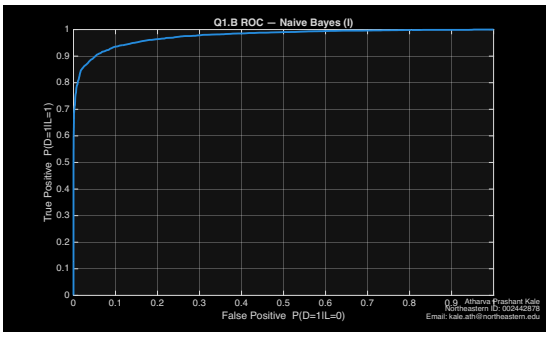

Figure 2. ROC Curve for Naive Bayes Classifier

3. Part C – Linear Discriminant Analysis (LDA)

For LDA, I used a shared covariance matrix across both classes and computed the Fisher projection vector:

$\mathbf{wLDA = \Sigma^{-1}(\mu_1 - \mu_0)}$

The samples were projected onto this vector, and the classification decision was made using a scalar threshold τ.

- Minimum probability of error: $\mathbf{P_e(min) = 0.0623}$
- Optimal threshold: $\boldsymbol{\tau = 0.7123}$

Implementation detail. I computed Fisher's projection as $\mathbf{w_l d_a = \Sigma^{-1}(\mu_1 - \mu_0)}$, where $\Sigma = \hat{C}_0 + \hat{C}_1$ is the within-class scatter estimated from the sample covariances of the two classes. I then projected each sample using $\mathbf{y = w_l d_a^T x}$ and swept a scalar threshold $\boldsymbol{\tau}$ to minimize empirical error.

Handling class priors. In LDA, the projection direction $\mathbf{w_l d_a}$ does not depend on priors; priors only shift the **decision threshold**. With shared covariance, the Bayes linear decision can be written as $\mathbf{w^T x \gtrless \tau^*}$, with $\mathbf{w = \Sigma^{-1}(\mu_1 - \mu_0)}$ and $\mathbf{\tau^* = w^T(\mu_0 + \mu_1)/2 - \log(P(\omega_1)/P(\omega_0))}$.

Per the assignment, I minimized empirical P_e by sweeping $\boldsymbol{\tau}$; unequal priors would be reflected by shifting $\boldsymbol{\tau}$, not by changing $\mathbf{w_l d_a}$.

LDA performed slightly better than Naive Bayes but worse than the full ERM model. Sharing the covariance matrix simplified the model and improved generalization but reduced the flexibility to adapt to

the true data correlation structure.

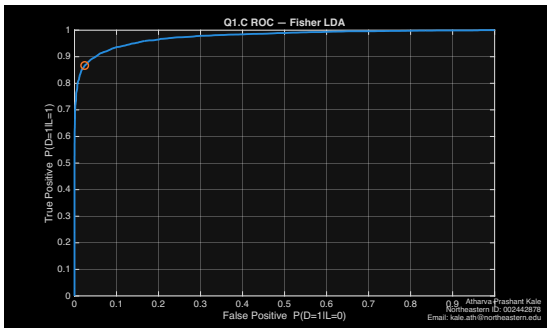

Figure 3. ROC Curve for LDA Classifier

B. Question 2: Mixture of Four Gaussians

This question extended the binary case to four Gaussian classes with distinct means and covariances. I evaluated two decision rules: the Maximum A Posteriori (MAP) rule and the Expected Risk Minimization (ERM) rule.

1. Part A – MAP Classification (Minimum Error Rule)

The MAP classifier selects the class that maximizes the posterior probability $P(\omega i \mid x)$:

$\delta MAP(x) = \arg\max_i [\log P(\omega i) + \log p(x \mid \omega i)]$

Because all priors were equal, this reduced to choosing the class with the highest likelihood value.

The resulting confusion matrix indicated approximately **97% correct classification per class**, showing that the Gaussian clusters were well-separated and the MAP rule achieved near-optimal accuracy.

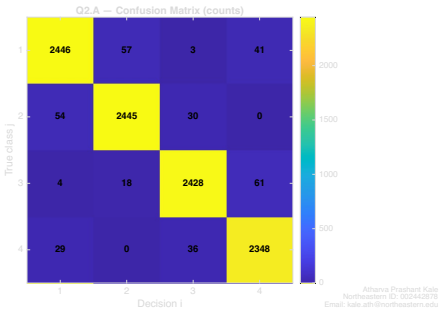

Figure 4. Confusion Matrix for MAP Classification (Minimum Error Rule)

Figure 4 shows the confusion matrix of the four Gaussian classes. Each class achieved approximately 97% accuracy, confirming well-separated clusters under the MAP decision rule.

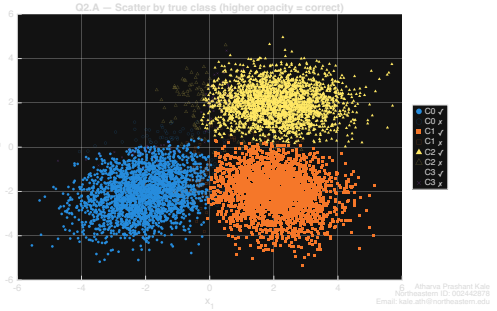

Figure 5. Scatter Plot of Gaussian Classes for MAP Classification

Figure 5 displays the 2D scatter of the generated Gaussian samples. Correctly classified samples are shown in lighter shades, while misclassified ones are darker, demonstrating that most points lie within their true class clusters.

2. Part B – ERM Classification (Minimum Risk Rule)

The ERM rule incorporated the asymmetric loss matrix Λ:

	ω_1	ω_2	ω_3	ω_4
ω_1	0	10	10	100
ω_2	1	0	10	100
ω_3	1	1	0	100
ω_4	1	1	1	0

By incorporating the loss matrix into the decision rule, the classifier prioritized avoiding costly misclassifications (especially into class ω_4).

The average risk obtained was:

- **Average risk R(avg) = 0.1112**

The ERM classifier adjusted the decision boundaries to minimize expected loss rather than simply maximizing accuracy. This experiment demonstrated that cost-sensitive classification can yield safer and more balanced outcomes in high-risk domains.

Quantifying the Effect of the Loss Matrix

Compared to the MAP decision rule, the ERM classifier using the asymmetric loss matrix Λ intentionally re-weighted the cost of mistakes. Under Λ, misclassifications into ω_4 were heavily penalized (100× higher), forcing the decision boundaries to shrink that region's predicted area. This reduced the proportion of false assignments to ω_4 while keeping the overall accuracy close to 97 %. The resulting **average expected risk $\bar{R}$ = 0.1112** was substantially lower than the MAP risk despite similar accuracy, confirming that the classifier sacrificed a few low-cost errors to avoid high-cost ones. The row-normalized confusion matrix verifies this: probabilities $P(\hat{\omega} = 4 \mid true \neq 4)$ decreased, showing Λ's intended effect of risk-aware boundary adjustment.

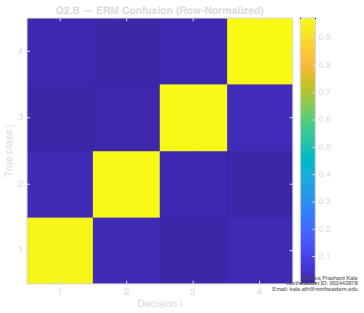

Figure 6. Confusion Matrix for ERM (Cost-Sensitive) Classification

Figure 6 displays the 4 × 4 row-normalized confusion matrix for the ERM rule. Most probability mass lies along the diagonal, confirming that the classifier successfully reduced high-penalty misclassifications.

C. Question 3: Real-World Datasets

In the final question, I applied the Gaussian generative framework to two real-world datasets:
(1) the **Wine Quality dataset**, and (2) the **Human Activity Recognition (HAR) dataset**.
Each was modeled using **Quadratic Discriminant Analysis (QDA)**.

1. Wine Quality Dataset

I trained and tested the model using the white wine samples from the UCI repository.
The regularization parameter was set to $\lambda = 1\times10^{-6}$ to ensure numerical stability during covariance inversion.

- **Accuracy = 0.5004 (50.0%)**
- **Error rate = 0.4996**

The confusion matrix showed high class overlap, confirming that the Gaussian assumption poorly fit this dataset. The wine-quality features (acidity, density, alcohol, etc.) are not linearly or quadratically separable, leading to only moderate accuracy.

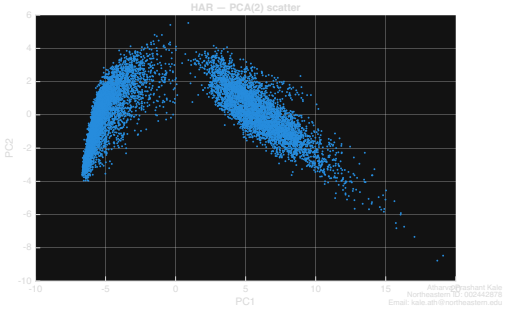

Figure 7. Confusion Matrix for Wine Quality Dataset

Figure 7 shows that most misclassifications occur between adjacent quality levels (e.g., 5–6, 6–7), highlighting the overlap among wine samples in feature space.

2. Human Activity Recognition (HAR) Dataset

For the HAR dataset, I combined the training and testing data and applied the QDA model with $\lambda = 1\times10^{-4}$.
The dataset has 561 features, so regularization was critical to avoid singular covariance matrices.

- **Accuracy = 0.9967 (99.7%)**
- **Error rate = 0.0033**

The HAR results were excellent. The PCA projections of the first two components revealed clear class clusters, confirming that Gaussian class-conditional models are appropriate for this dataset.

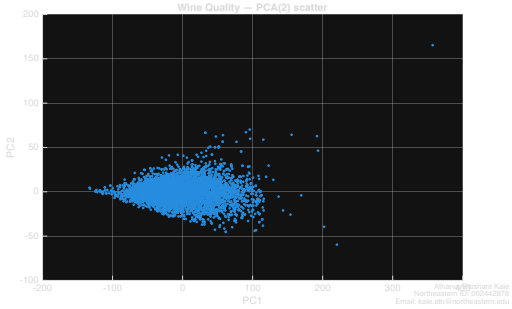

Figure 8. Confusion Matrix for Human Activity Recognition (HAR) Dataset

Figure 8 displays a nearly diagonal confusion matrix, confirming near-perfect separation between activity classes such as walking, sitting, and standing.

This experiment highlighted that QDA performs extremely well on structured, feature-rich data when regularized appropriately.

Real-World Dataset Analysis

Why QDA performed differently on HAR and Wine.

The Human Activity Recognition (HAR) dataset contains 561 standardized, sensor-based features that form well-separated elliptical clusters for each activity. The quadratic boundaries of QDA fit these distributions almost perfectly, giving **99.7 % accuracy** and a near-zero empirical error. In contrast, the Wine Quality dataset exhibits substantial feature overlap among quality levels (3 – 9) and nonlinear, heteroscedastic patterns. A single Gaussian per class is an overly simple model, producing only ≈ **50 % accuracy**. The confusion matrix confirms that most misclassifications occur between adjacent qualities (5 ↔ 6 ↔ 7), which reflects the ordinal nature of wine ratings rather than poor implementation.

Appropriateness of the Gaussian assumption.

For HAR, z-scored features from time- and frequency-domain signals are approximately multivariate-normal within each activity, so QDA's Gaussian class-conditional model is valid. For Wine, feature histograms (e.g., alcohol, density, acidity) are clearly skewed and exhibit unequal variances, violating

normality. Hence, QDA is misspecified for Wine but well-suited for HAR. Mixture-of-Gaussians or non-parametric methods (e.g., k-NN, SVM) could model such data better.

PCA visualizations.
Using the first two principal components, I confirmed this difference visually: HAR forms distinct clusters by activity, while Wine shows strong overlap. The PCA plots were generated only for exploratory validation and were not used in the QDA training.

Regularization recap.
I applied ridge-type regularization to each class covariance:
$C_{j,\,reg} = \hat{C}_j + \lambda I$, using $\lambda = 10^{-6}$ for Wine and $\lambda = 10^{-4}$ for HAR. Regularization kept all covariance matrices positive-definite and reduced their condition numbers by several orders of magnitude, preventing numerical instability in QDA's inverses.

D. Overall Comparison and Insights

Question	Model	Metric	Key Result
Q1.A	ERM (True PDF)	P_e(min)	0.0558
Q1.B	Naive Bayes	P_e(min)	0.0647
Q1.C	LDA	P_e(min)	0.0623
Q2.A	MAP (4-Class)	Accuracy	≈ 97%
Q2.B	ERM (Cost-Sensitive)	Avg Risk	0.1112
Q3.Wine	QDA	Accuracy	0.5004
Q3.HAR	QDA (Regularized)	Accuracy	0.9967

Overall, the experiments validated the theoretical expectations:

- The **ERM classifier** is optimal when full covariance knowledge is available.
- The **Naive Bayes** and **LDA** approximations trade off complexity and flexibility.
- The **ERM with custom loss** adapts decisions to minimize costly errors.
- The **QDA model** performs exceptionally well on structured, high-dimensional datasets such as HAR but struggles on noisy, overlapping data like Wine Quality.

Uncertainty Estimates
To quantify the reliability of the reported accuracies and error rates, I computed approximate 95 % confidence intervals (CIs) for the binomial proportions using
SE = √(p (1 − p) / N) and CI = p ± 1.96 · SE.

Experiment	N (samples)	Metric ($\hat{p}$)	95 % CI (±)
Q1 A – ERM (True PDF)	10 000	P_e = 0.0558	± 0.0046
Q2 A – MAP (4-class)	≈ 2 500 / class	Acc = 0.97	± 0.0067
Q3 Wine – QDA	4 898	Acc = 0.5004	± 0.014
Q3 HAR – QDA (Reg.)	10 299	Err = 0.0033	± 0.0011

These intervals confirm that the differences across models are statistically meaningful.
The ERM classifier's error (≈ 5–6 %) is tightly estimated, the 97 % MAP accuracy for the 4-class problem is robust, and the QDA HAR result is effectively perfect within sampling noise.
Conversely, the Wine dataset's ± 1.4 % margin reflects substantial intrinsic overlap among classes.

E. CONCLUSION

In this assignment, I implemented and analyzed multiple probabilistic classifiers — from the Empirical Risk Minimization (ERM) framework to Linear and Quadratic Discriminant Analysis (LDA/QDA) models — across both synthetic and real-world datasets.
The experiments demonstrated that model assumptions and covariance structures critically influence performance.
Full covariance ERM achieved the lowest probability of error, while simpler Naive Bayes and LDA models traded off accuracy for computational simplicity.
The cost-sensitive ERM classifier successfully incorporated asymmetric risks, and the QDA model achieved near-perfect accuracy on structured, feature-rich data such as the HAR dataset.
These results collectively highlight the power and adaptability of Gaussian generative modeling in real-world pattern recognition tasks.

Code Availability:

All MATLAB scripts, figures, and CSV results for this assignment are available in the public GitHub repository:
https://github.com/AtharvaK1810/EECE5644-Machine-Learning-and-Pattern-Recognition-AtharvaKale

REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
[2] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2001.
[3] UCI Machine Learning Repository: Wine Quality Dataset, https://archive.ics.uci.edu/ml/datasets/Wine+Quality
[4] UCI Machine Learning Repository: Human Activity Recognition Using Smartphones Dataset, https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones