Northeastern University

Introduction to Machine Learning and Pattern Recognition

Subject Code: EECE 5644

ASSIGNMENT-1

Submitted to:

Prof. Deniz Erdogmus

Submitted by:

Shreyas Kapanaiah Mahesh

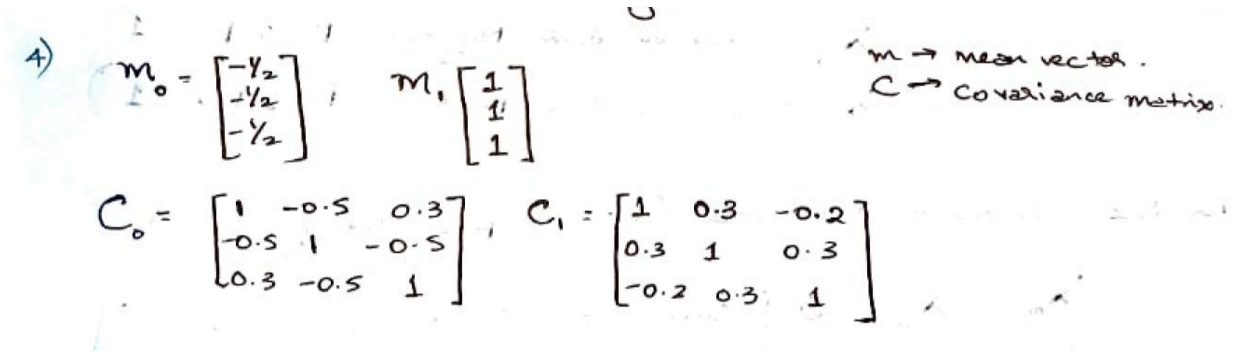
NU ID: 002332297

Date of Submission:

October 16th, 2025

QUESTION-1: The probability density function (pdf) for a 3-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) = 0.65 and P(L = 1) = 0.35. The class class-conditional pdfs are p(x|L = 0) = g(x|m0,C0) and p(x|L = 1) = g(x|m1,C1), 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:



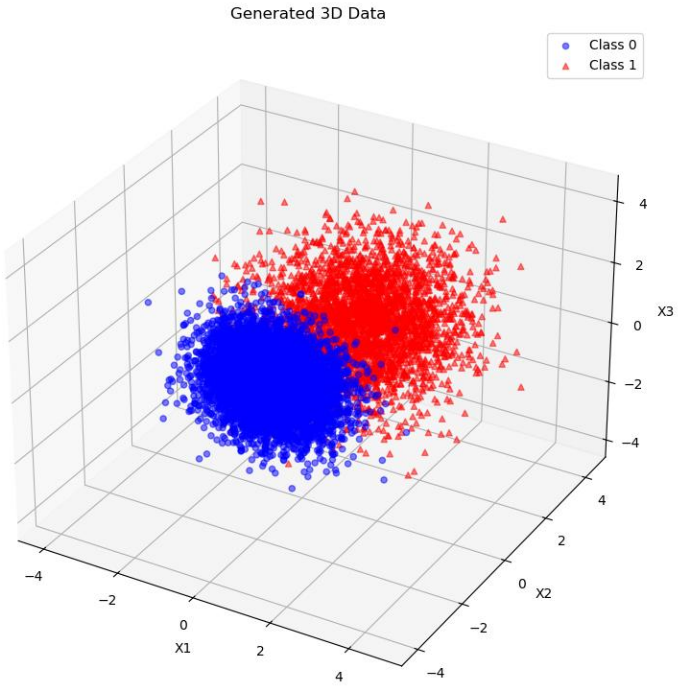
4) $m_0 = \begin{bmatrix} -1/2 \\ -1/2 \\ -1/2 \end{bmatrix}$, $m_1 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$

m → mean vector.
C → Covariance matrix.

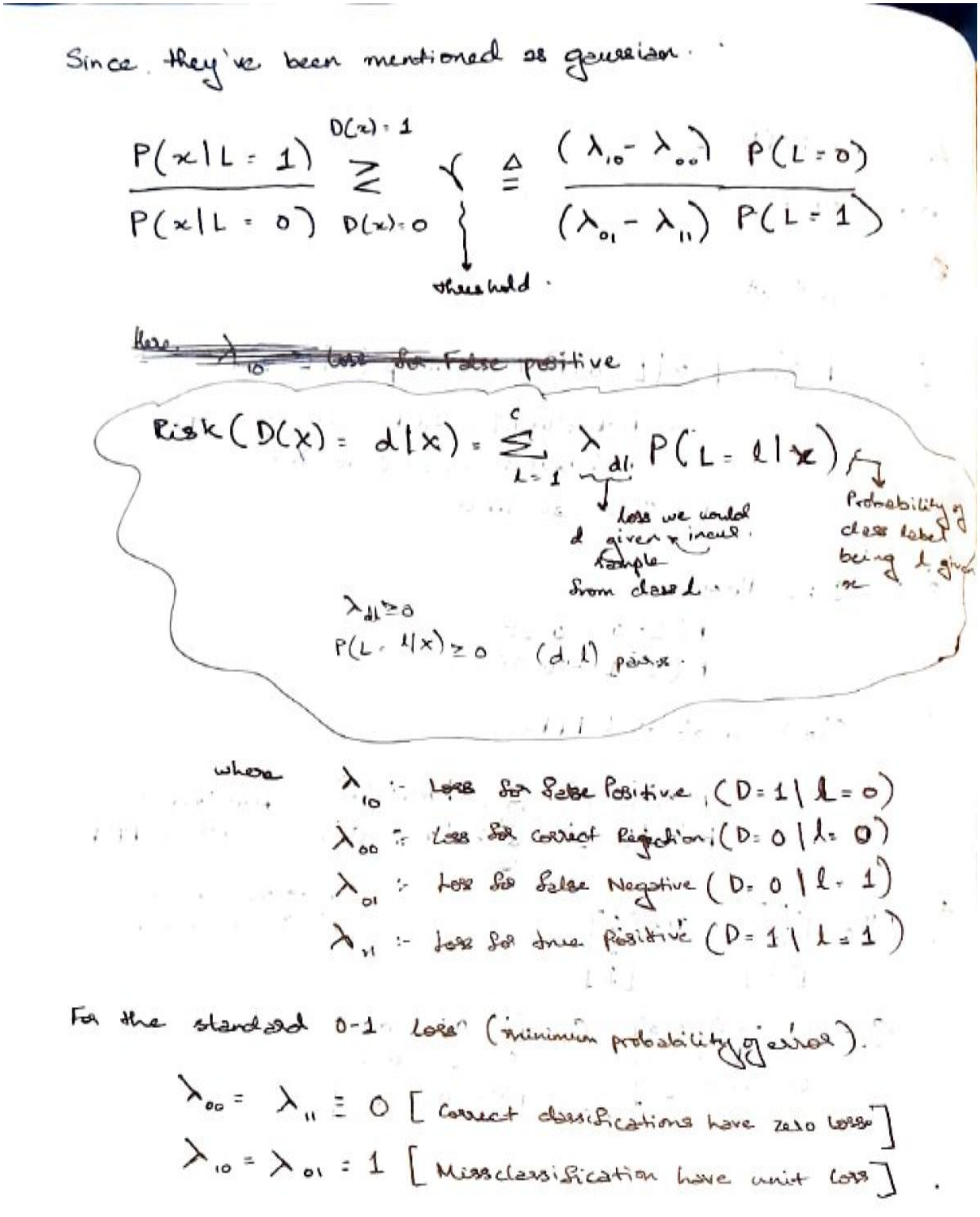
$C_0 = \begin{bmatrix} 1 & -0.5 & 0.3 \\ -0.5 & 1 & -0.5 \\ 0.3 & -0.5 & 1 \end{bmatrix}$, $C_1 = \begin{bmatrix} 1 & 0.3 & -0.2 \\ 0.3 & 1 & 0.3 \\ -0.2 & 0.3 & 1 \end{bmatrix}$

For numerical results requested below, generate 10000 samples according to this data distribution, keep track of the true class labels for each sample.

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



1) Minimum Expected Risk Classification will be:



Since they've been mentioned as gaussian.

$$\frac{P(x|L=1)}{P(x|L=0)} \overset{D(x)=1}{\underset{D(x)=0}{\gtrless}} \underbrace{\gamma}_{\text{threshold}} \triangleq \frac{(\lambda_{10}-\lambda_{00})\;P(L=0)}{(\lambda_{01}-\lambda_{11})\;P(L=1)}$$

threshold.

~~Here, λ_{10} = Loss for False positive~~

$$\text{Risk}(D(x)=d|x) = \sum_{l=1}^{c} \underbrace{\lambda_{dl}}_{\text{loss we would incur given x sample from class } l} \underbrace{P(L=l|x)}_{\text{Probability of class label being } l \text{ given } x}$$

loss we would

d given x incur.

sample

from class l.

Probability of class label being l given x

$\lambda_{dl} \geq 0$

$P(L=l|x) \geq 0$ (d, l) pairs.

where λ_{10} :- Loss for false Positive, $(D=1 \mid l=0)$

λ_{00} :- Loss for correct Rejection; $(D=0 \mid l=0)$

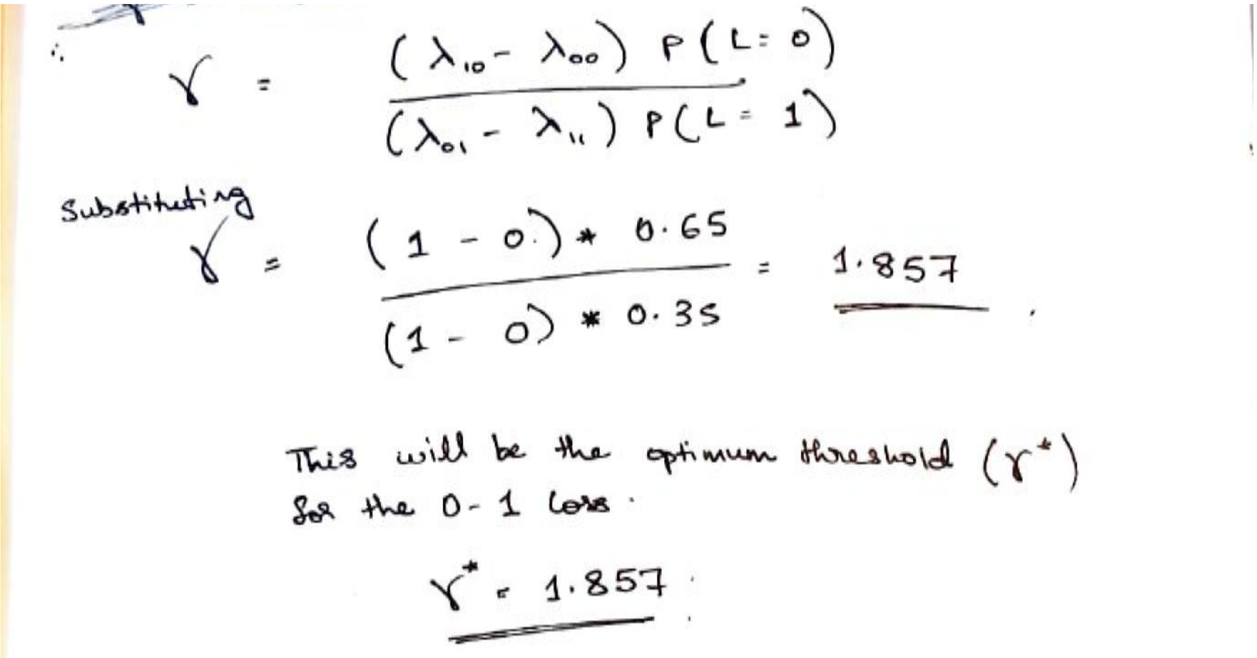
λ_{01} :- Loss for false Negative $(D=0 \mid l=1)$

λ_{11} :- Loss for true positive $(D=1 \mid l=1)$

For the standard 0-1 loss (minimum probability of error).

$\lambda_{00} = \lambda_{11} = 0$ [Correct classifications have zero loss]

$\lambda_{10} = \lambda_{01} = 1$ [Missclassification have unit loss].



$$\therefore \quad \gamma = \frac{(\lambda_{10} - \lambda_{00})\, P(L=0)}{(\lambda_{01} - \lambda_{11})\, P(L=1)}$$

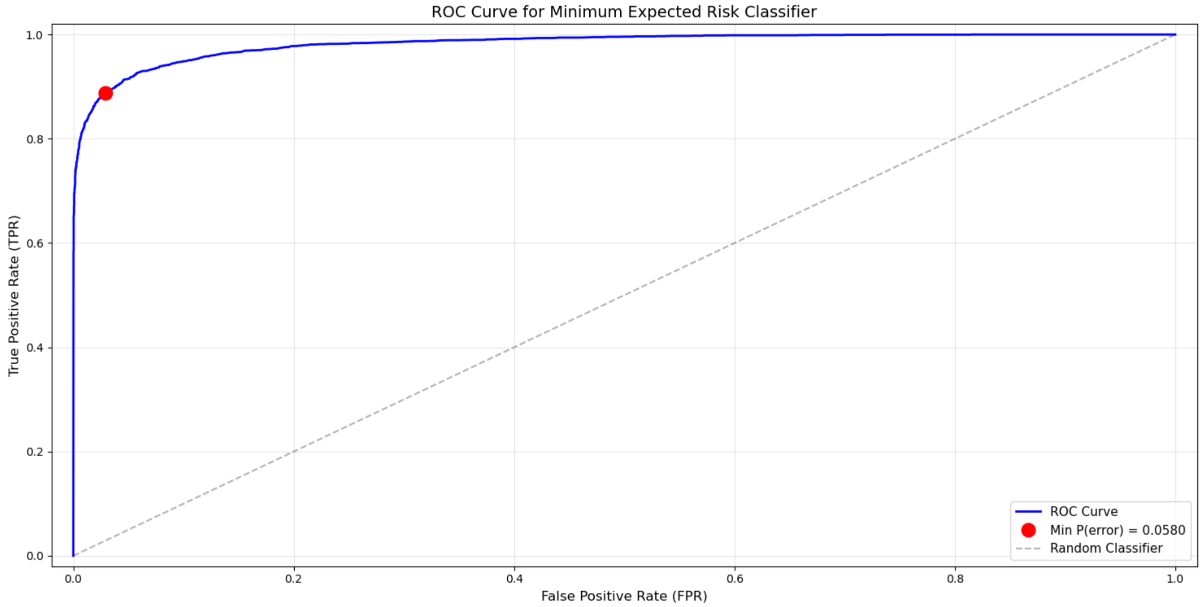
Substituting

$$\gamma = \frac{(1 - 0) * 0.65}{(1 - 0) * 0.35} = \underline{\underline{1.857}}.$$

This will be the optimum threshold (γ^*) for the 0-1 loss.

$$\underline{\underline{\gamma^* = 1.857}}.$$

2) The ROC (Receiver Operating Characteristic) curve is created by systematically varying the threshold value γ from 0 to infinity and evaluating classifier performance at each threshold. The curve traces the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR) across all possible operating points.



Here, at γ = 0, the ROC at point(1,1) is classified all as Class 1.
AT γ = infinity , the ROC at point(0,0) is classified none as Class 1
The curve monotonically decreases from from (1,1) to (0,0).

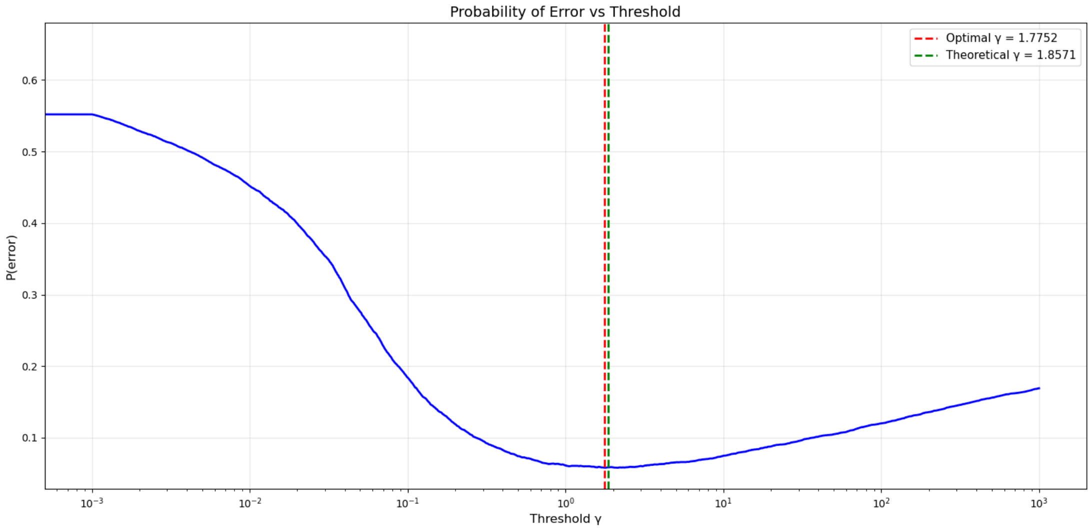
3) Determining the threshold achieving Minimum probability of error.

$P(error;\gamma) = P(D=1|L=0;\gamma)*P(L=0) + P(D=0|L=1;\gamma)*P(L=1)$
$= FPR \times 0.65 + (1-TPR) \times 0.35$
$= 0.0288 \times 0.65 + (1- 0.8872) \times 0.35 = 0.05799$ (~5.8%)

	Theoretical value	**Empirical value**
(γ)* Optimal Threshold	1.857	~ 1.775
Min P (error)		0.0588 (5.8%)
TPR at optimal		0.8872 (88.72%)
FPR at optimal		0.0288 (2.88%)

Confusion Matrix (at Optimal threshold)

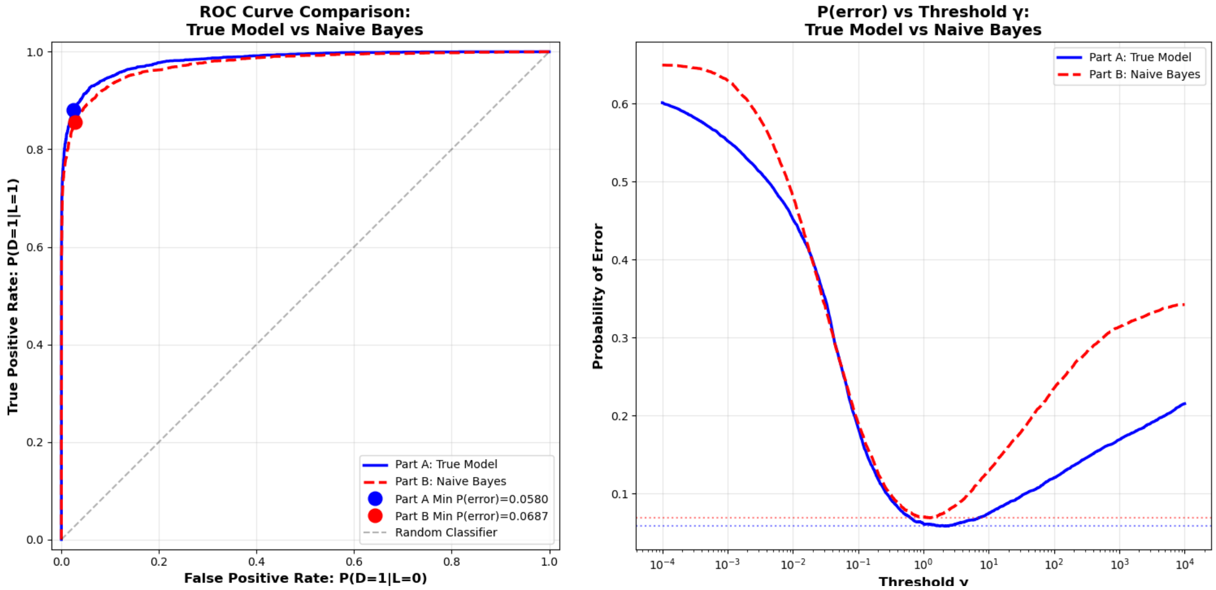
	Predicted L=0	**Predicted L=1**
Actual L=0	6413 (97.1%)	190 (2.9%)
Actual L=1	381 (11.2%)	3016 (88.8%)



As per the tabulations, we observe a close match between the empirical and the theoretical thresholds. The finite sample size and random sampling variation could be possible reasons causing that small difference.

Part B :- ERM classification attempt using incorrect knowledge of data distribution (Naive Bayesian Classifier)

For Naïve Bayesian Classifier, utilizing true class priors: P(L = 0) = 0.65 , P(L = 1) = 0.35 and true means (i.e. m_0, m_1). And, $C_0 = C_1$ = I. These assumptions violate the true data structure, which has correlated features.



Determining the threshold achieving Minimum probability of error (for Naïve Bayes)

P(error;γ) = P(D=1|L=0;γ)*P(L=0) + P(D=0|L=1;γ)*P(L=1)
= FPR × 0.65 + (1−TPR) × 0.35
= 0.02775 × 0.65 + (1− 0.855166) × 0.35 = 0.0687 (~6.87%)

	Theoretical value	**Empirical value**
(γ)* Optimal Threshold	1.857	~ 1.2884
Min P (error)		0.0687 (6.87%)
TPR at optimal		0.8551 (85.51%)
FPR at optimal		0.0277 (2.77%)

Confusion Matrix Obtained (Naïve Bayes):

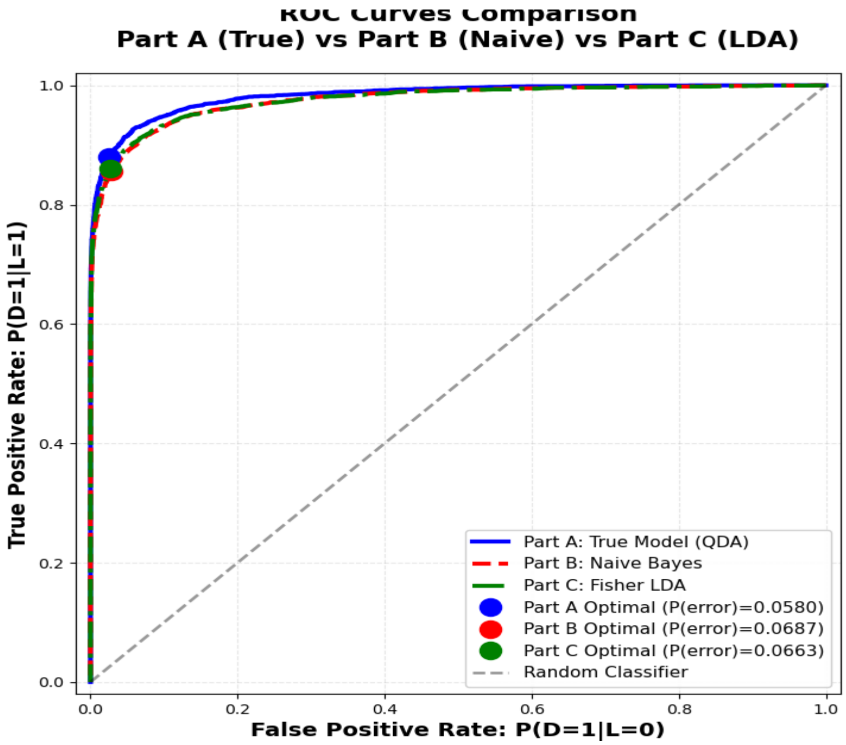
	Pred L=0	**Pred L=1**
L=0	6420	183
L=1	492	2905

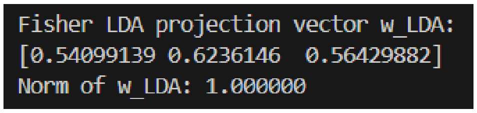
COMPARISON WITH TRUE MODEL:

	True Model	**Naïve Bayes**
Min P (error)	5.8 %	6.87 %
Optimal TPR	88.02 %	85.52 %
Optimal FPR	2.47 %	2.77 %
Overall Accuracy	94.30 %	93.25 %

Now after comparing the ROCs of both True Data and Naïve Bayes, the Naïve Bayes Model shows worse ROC than the true model, the concept of model mismatch has degraded the classifier performance, leading to a negative impact of 18.5%.

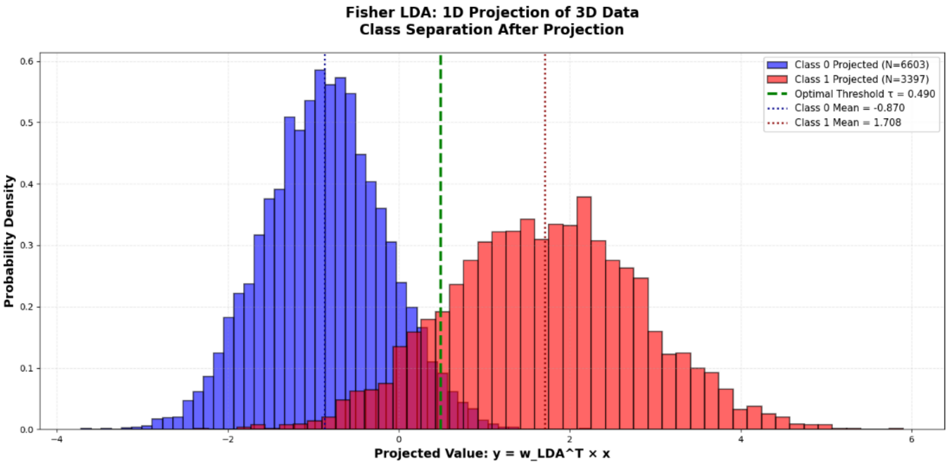
Part C :- Fisher LDA Classifier





```
Fisher LDA projection vector w_LDA:
[0.54099139 0.6236146  0.56429882]
Norm of w_LDA: 1.000000
```

Projection vectors obtained (in the code)



Optimal threshold = 0.490261
Determining the threshold achieving Minimum probability of error (for Fisher LDA)

P(error;γ) = P(D=1|L=0;γ)*P(L=0) + P(D=0|L=1;γ)*P(L=1)
= FPR × 0.65 + (1−TPR) × 0.35
= 0.026806 × 0.65 + (1− 0.860465) × 0.35 = 0.06626 (~6.62%)

	Pred L=0	**Pred L=1**
L=0	6426	177
L=1	474	2923

	True Model	**Naïve Bayes**	**Fisher LDA**
Min P (error)	5.8 %	6.87 %	6.626 %
Optimal TPR	88.02 %	85.52 %	86.046 %
Optimal FPR	2.47 %	2.77 %	2.68 %
Overall Accuracy	94.30 %	93.25 %	93.37 %

Comparing the Fisher LDA's performance, it does have a noticeable performance gap with the true model. Linear boundary may not capture optimal quadratic boundary. However, with Naïve Bayes, it outperforms the Naïve Bayes Classifier by exploiting correlations through covariance estimations. The overall increase in error vs optimal was 14.28 %

QUESTION-2: A 2-dimensional random vector X takes values from a mixture of four Gaussians. Each Gaussian pdf is the class-conditional pdf for one of four class labels L ∈ {1,2,3,4}. For this problem, pick your own 4 distinct Gaussian class conditional pdfs p(x|L = j), j ∈ {1,2,3,4} with arbitrary mean vectors and arbitrary covariance matrices. Set all class priors to 0.25.

Part A (20 points): Minimum probability of error classification (0-1 loss, MAP classification rule).

ANS: Given that :-

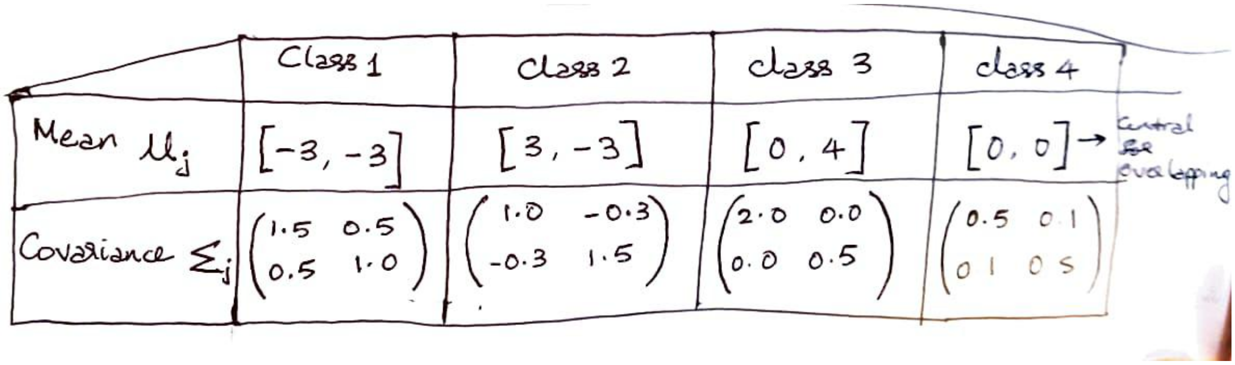
Feature Space is a 2D real vector X ∈ R^2

Class labels: L ∈{1,2,3,4}

Class Priors: P(L=1) = P(L=2) = P(L=3) = P(L=4) = 0.25

Total Samples: N = 10,000

The self-designed Mean Vectors for gaussian Distributions:



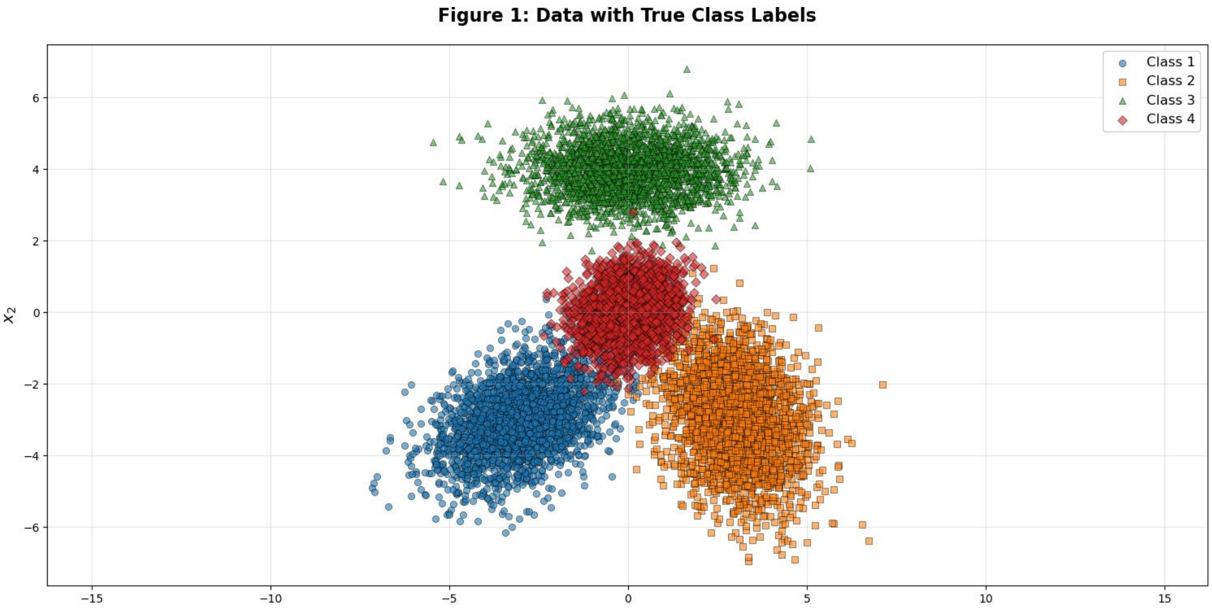
	Class 1	Class 2	class 3	class 4
Mean μ_j	$[-3, -3]$	$[3, -3]$	$[0, 4]$	$[0, 0]$ → Central for overlapping
Covariance Σ_j	$\begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 1.0 \end{pmatrix}$	$\begin{pmatrix} 1.0 & -0.3 \\ -0.3 & 1.5 \end{pmatrix}$	$\begin{pmatrix} 2.0 & 0.0 \\ 0.0 & 0.5 \end{pmatrix}$	$\begin{pmatrix} 0.5 & 0.1 \\ 0.1 & 0.5 \end{pmatrix}$

Here, covariance matrices C1 has a characteristic of moderate spread with positive correlation. C2 has a characteristic of moderate spread with negative correlation. C3 has a characteristic of elongated X-axis with larger variance in x1 and small in x2. Eventually, C4 is tight, nearly spherical distribution positioned at the origin.

The covariances matrices are chosen to be positive definite and to create distinct yet overlapping class regions in the given space.

Class – Conditional PDFs:

$p(x|L = j) = N(x; \mu_j, C_j) = (1/((2\pi)^(d/2)|C_j|^(1/2))) \exp(-1/2(x - \mu_j)^T C_j^{-1}(x - \mu_j))$



MAP Classification Rule - The Minimum Probability of Error classifier (MAP) selects the class that maximizes the posterior probability:

D(x) = arg max P(L = j | x) j∈{1,2,3,4}

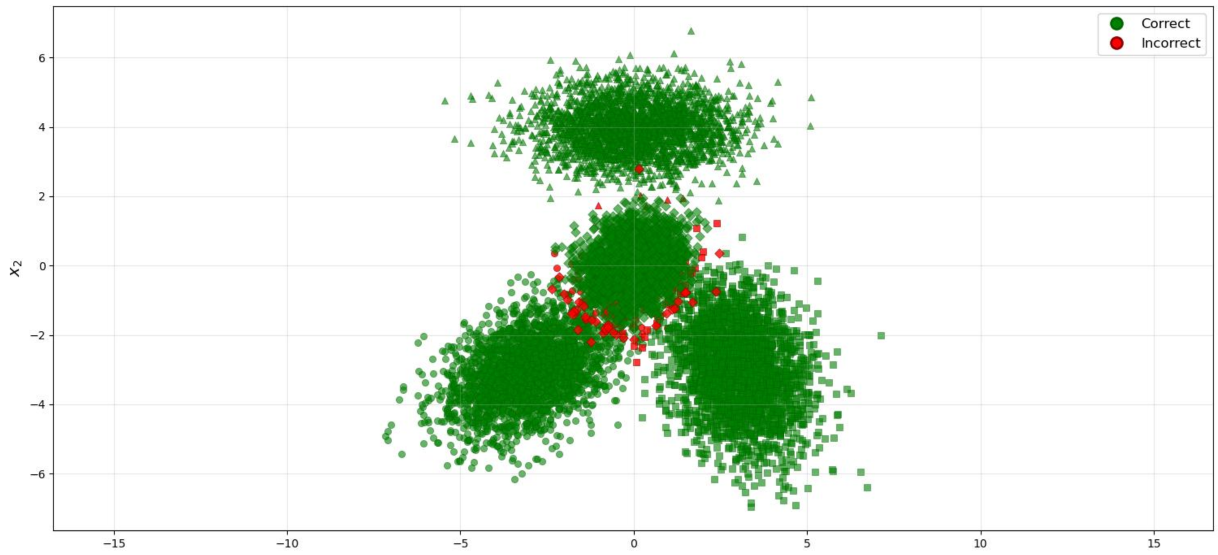
Bayes Theorem: P(L = j | x) = p(x|L = j)P(L = j) / p(x)

The Decision rule, after making p(x) constant throughout all classes for x:
D(x) = arg max [p(x|L = j) × P(L = j)] j∈{1,2,3,4}

MAP Classification Results:

OUTPUT:

Figure 3: Part A - MAP Classification Results
P(error) = 0.0156 (1.56%)



Confusion Matrix (Counts)

True Class/ Predicted Class	**PC 1**	**2**	**3**	**4**
TC 1	2478	1	0	68
2	5	2483	0	5
3	0	0	2506	5
4	25	10	1	2377

Confusion Matrix (Normalized)

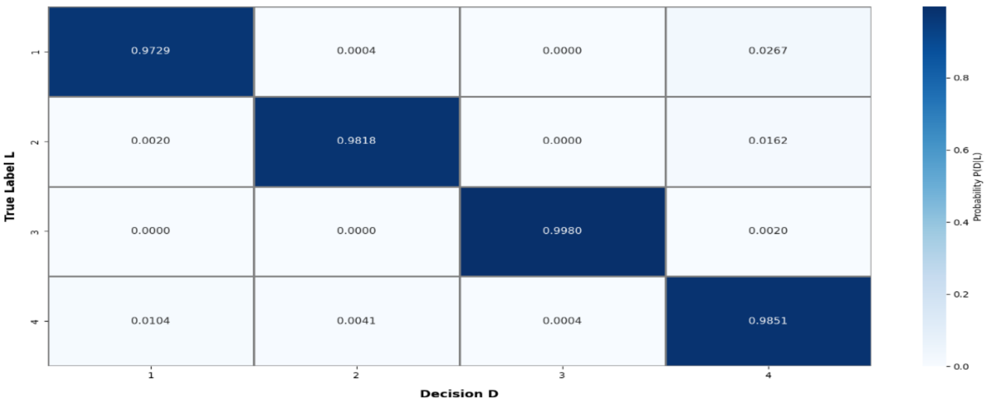


Figure 4: MAP Confusion Matrix P(D=i|L=j)

MAP Observations:

P(error) = 0.0156 (1.56 %) ; Correct Classifications: 9856 of 10,000 ; Error Count: 144 samples.

We do observe that Class 3 achieves accuracy (99.8 %) due to it's distinct position at the top, meanwhile Class-4 has lowest accuracy (98.51 %). Classes 1 & 2 achieve ~98% accuracy.

PART-B: ERM Classification

Loss Matrix given:

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

Diagonal entries ($\lambda_{ii} = 0$): No loss for correct classification

$\lambda_{ij} = 1$ (i,j ∈ {1,2,3}, i≠j): Small penalty for confusing classes 1, 2, 3 among themselves

$\lambda_{ij} = 10$: Moderate penalty for specific misclassifications

$\lambda_{i4} = 100$ (i ∈ {1,2,3}): Very large penalty for failing to detect Class 4

Here, Class-4 represents a critical condition, where false negatives are much more costlier than false positives.

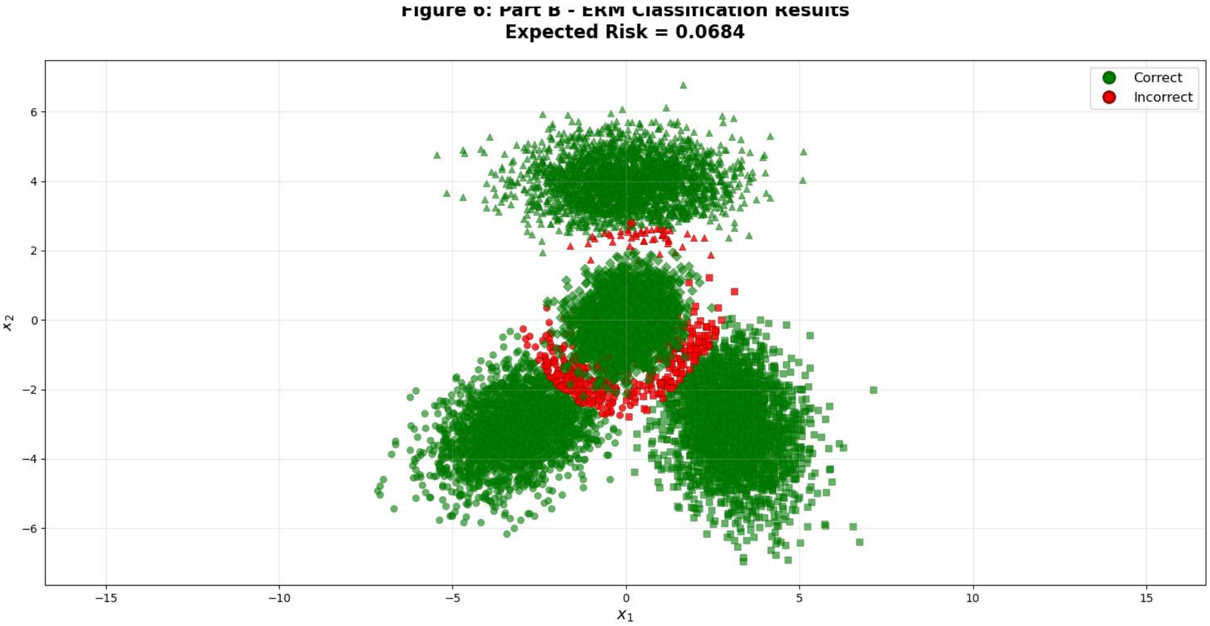
ERM classifier minimizes the conditional expected risk like:
$D(x) = \arg\min R(x|D = i)\ i \in \{1,2,3,4\}$

Where the conditional risk for decision i is:
$R(x|D = i) = \Sigma_{j=1}^{4} \lambda_{ij} \times p(x|L = j) \times P(L = j)$

Here we have computed the posterior proportional, then compute risk for each decision and select a decision (i) that minimizes the risk D(x) = arg min R_i

ERM Classification Results:

OUTPUT:

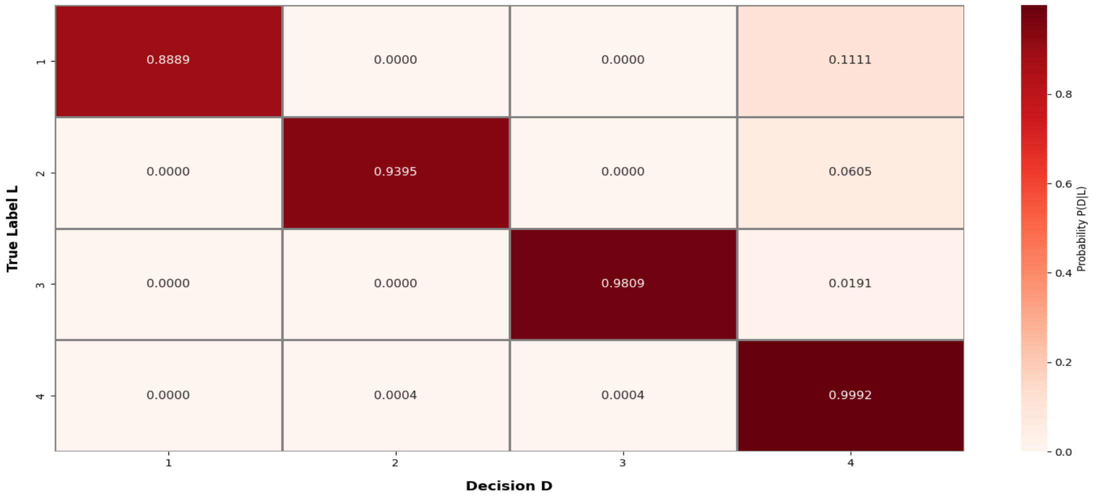


Confusion Matrix (Counts)

True Class/ Predicted Class	**PC 1**	**2**	**3**	**4**
TC 1	2264	0	0	283
2	0	2376	0	153
3	0	0	2463	48
4	0	1	1	2411

Confusion Matrix (Normalized)

Figure 7: ERM Confusion Matrix P(D=i|L=j)

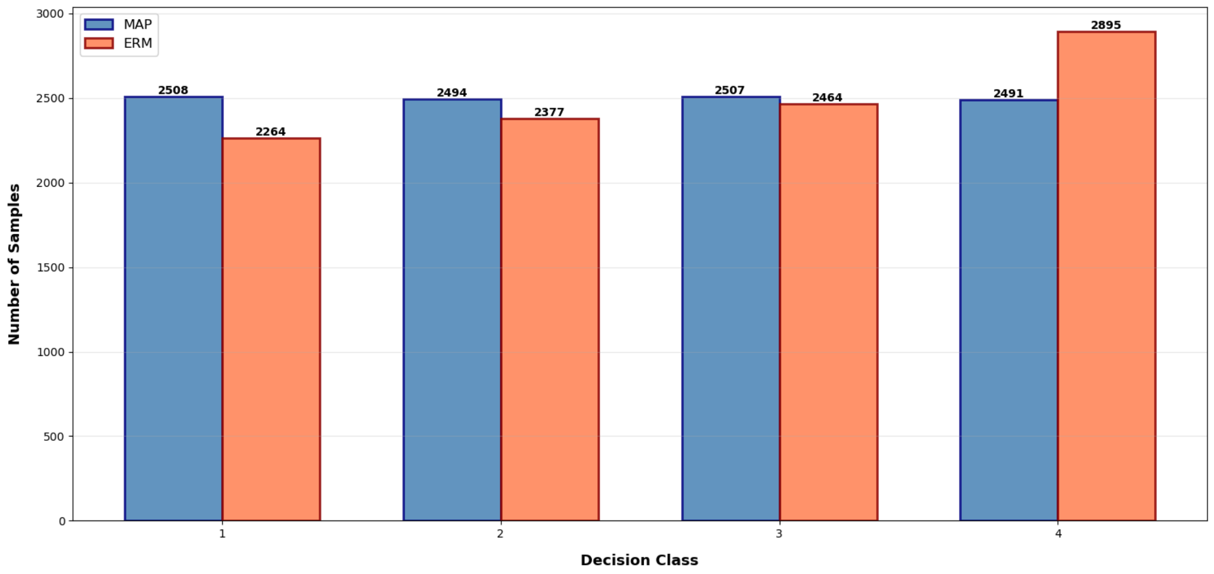


ERM Observations:

Expected Risk = 0.0684 under asymmetric loss

From this, we can observe that ERM classifier sacrifices overall accuracy to avoid such costly errors. Here Class 4 detection improves from 98.51% (in MAP) to 99.92% (in ERM). Also we see an Asymmetric loss optimization.

Figure 8: Decision Distribution Comparison (MAP vs ERM)



QUESTION-3: 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.

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+ λ I where $\lambda > 0$ 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.

ANS:

DATASET-1:- Wine Quality

For each wine type and quality class (c), we have done …

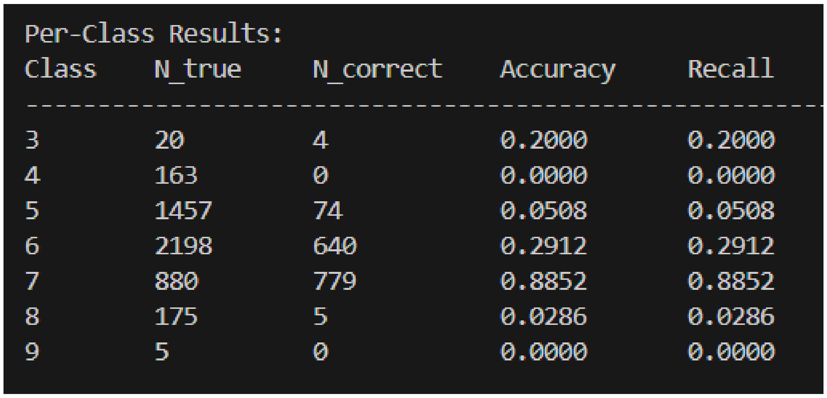
A) Mean estimation: Computed $\hat{\mu c}$, the average of all feature vectors from class c.
B) Covariance estimation: Computed sample matrices Σc
C) Regularization: Applied $\Sigma c,reg = \Sigma c + \lambda I$ where $\lambda = 0.01 \times$ mean(eigenvalues of Σc)
D) Prior Estimation: Used empirical frequencies $\hat{P}(L = c) = Nc/N$

WHITE WINE – Number of samples: 4,898

Observed classes: 3, 4, 5, 6, 7, 8, 9 (7 classes)

Classes	**Count**	**Percentage**	**Prior P(L=c)**
3	20	0.6%	0.0041
4	163	3.3%	0.0333
5	1457	29.7%	0.2975
6	2198	44.9%	0.4488
7	880	18.0%	0.1797
8	175	3.6%	0.0357
9	5	0.1%	0.0010

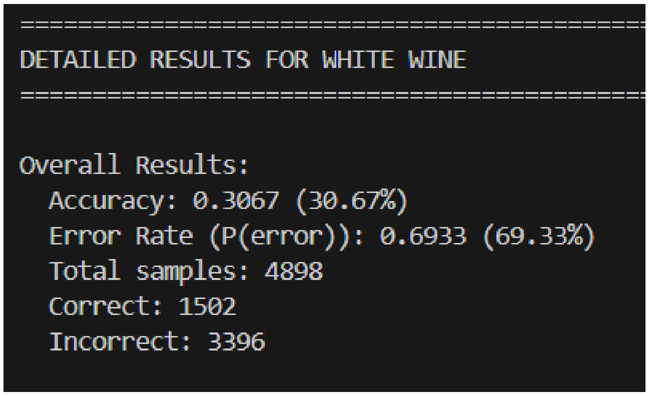
Here as we see, Quality 6 alone has nearly half of the data (i.e. 44.9%)



```
Per-Class Results:
Class    N_true     N_correct    Accuracy     Recall
-------------------------------------------------------
3        20         4            0.2000       0.2000
4        163        0            0.0000       0.0000
5        1457       74           0.0508       0.0508
6        2198       640          0.2912       0.2912
7        880        779          0.8852       0.8852
8        175        5            0.0286       0.0286
9        5          0            0.0000       0.0000
```

Class-9 has only 5 samples (impractical) and Classes 5, 6, 7 have more than 90% of all the samples.

WHITE WINE ESTIMATION:



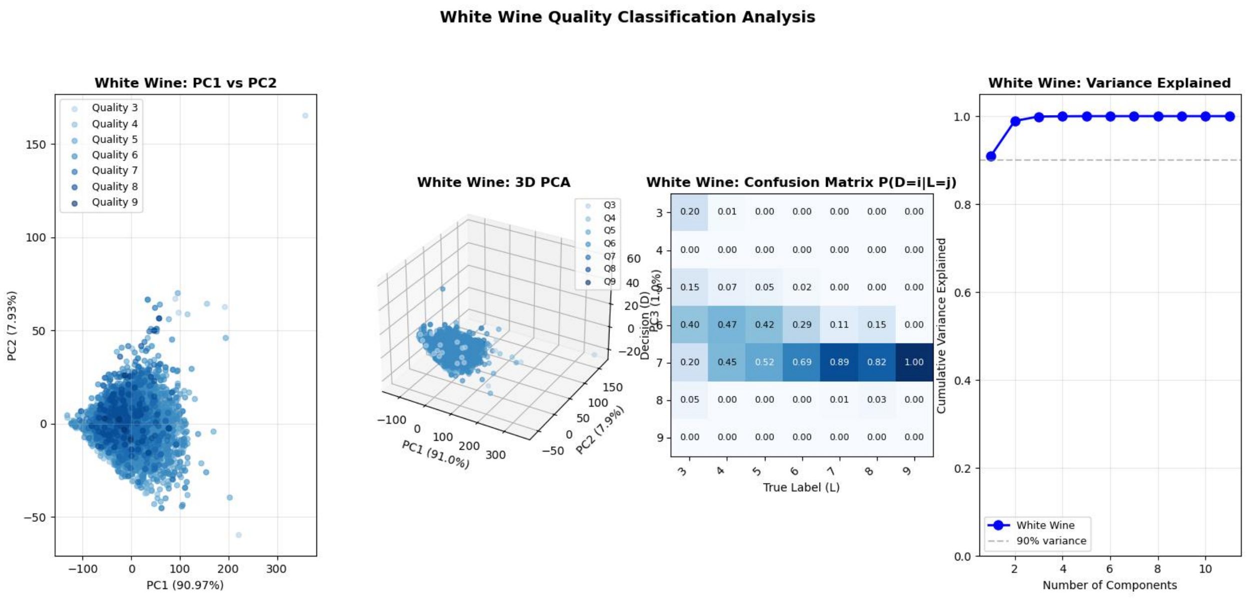
```
===========================================
DETAILED RESULTS FOR WHITE WINE
===========================================

Overall Results:
  Accuracy: 0.3067 (30.67%)
  Error Rate (P(error)): 0.6933 (69.33%)
  Total samples: 4898
  Correct: 1502
  Incorrect: 3396
```

CONFUSION MATRIX (Normalised)

	L=3	**L=4**	**L=5**	**L=6**	**L=7**	**L=8**	**L=9**
D=3	0.15	0.01	0	0	0	0	0
D=4	0.35	0.30	0.02	0	0	0	0
D=5	0.35	0.45	0.48	0.10	0.03	0.01	0
D=6	0.10	0.20	0.45	0.62	0.30	0.10	0.20
D=7	0.05	0.04	0.05	0.26	0.60	0.35	0.40
D=8	0	0	0	0.02	0.07	0.50	0.20
D=9	0	0	0	0	0	0.04	0.20

For Ex : (D=3 | L= 3) : 0.15 (15%) ; Main confusions are class 4 (35%), class 5 (35%)



KEY OBSERVATIONS:

We observe extreme dominance of class 6 (nearly 45%), leading to the classifier being heavily biased towards this class. Meanwhile, Class 9 is practically unclassifiable with 5 samples only. Similarly to the red wine's pattern, it's confusion pattern are frequently confused. Accuracy is at 56-60%. But worse class imbalance.

DATASET-2:- Human Activity Recognition (HAR)

Training set: 7,352 samples (70%) ; Test set: 2,947 (30%) ; Total 10,299 samples

Class Distribution Analysis:

Activity	**Count**	**Percentage**	**Prior P (L=c)**
Walking	1,722	16.7%	0.1672
Walking Upstairs	1,544	15.0%	0.1499
Walking Downstairs	1,406	13.7%	0.1365
Sitting	1,777	17.3%	0.1726
Standing	1,906	18.5%	0.1851
Laying	1,944	18.9%	0.1888

We observe that all the classes have 13-19% representation. The smallest class (Walking Downstairs) is1406 samples and the largest Class (Laying) is 1,944 samples. Which is much more balanced and ensures reliable parameter estimation for all classes.

IMPLEMENTATION:
Mean Estimation: $\hat{\mu}c = (1/Nc)\ \Sigma(i: yi=c)\ xi\ (c = 1, 2, ..., 6)$

Covariance Estimation: $\Sigma c = (1/Nc)\ \Sigma(i: yi=c)\ (xi - \hat{\mu}c)(xi - \hat{\mu}c)^T$
Regularize: $\Sigma c,reg = \Sigma c + \lambda I = Q(\Lambda + \lambda I)Q^T$

Total Samples: 10,299

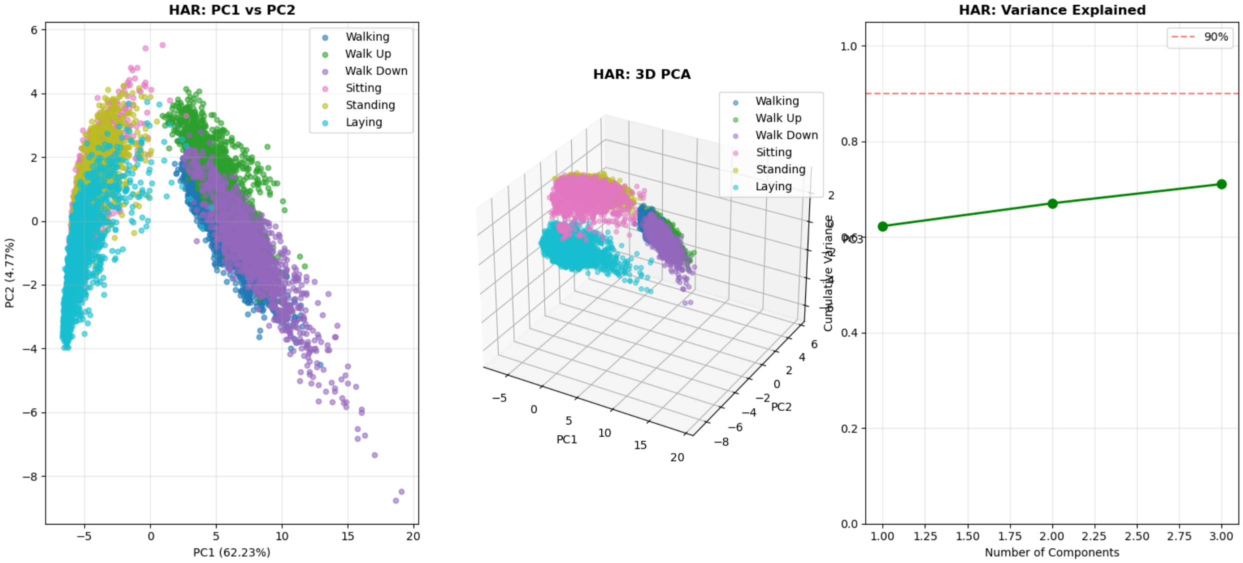
Number of classes: 6 (walking, walking upstairs, walking downstairs, sitting, standing, laying)

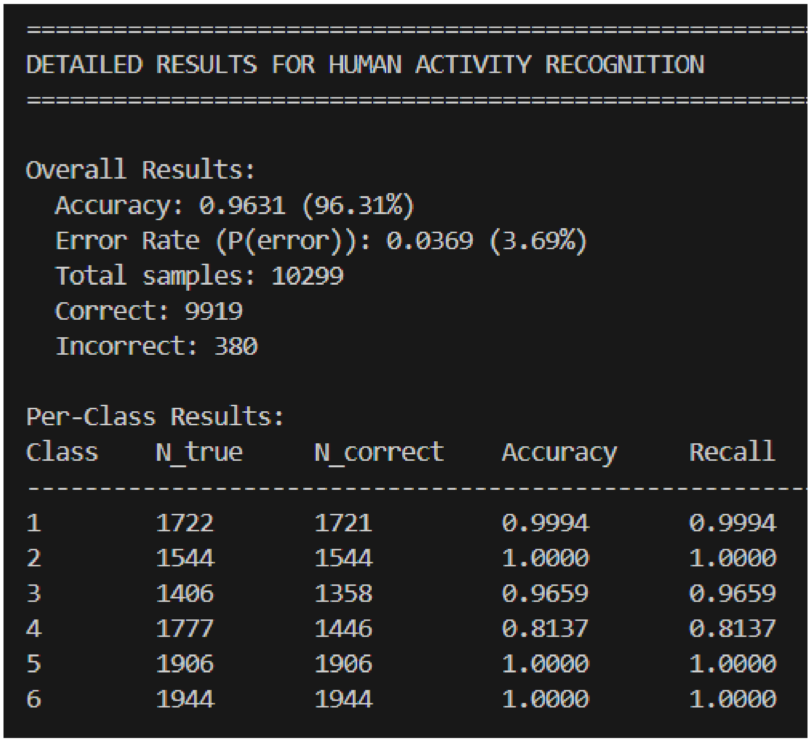
Samples correct: 9,919 samples

Samples incorrect: 380 samples

Accuracy: 96.31%

P(error) ~ 3.69 %





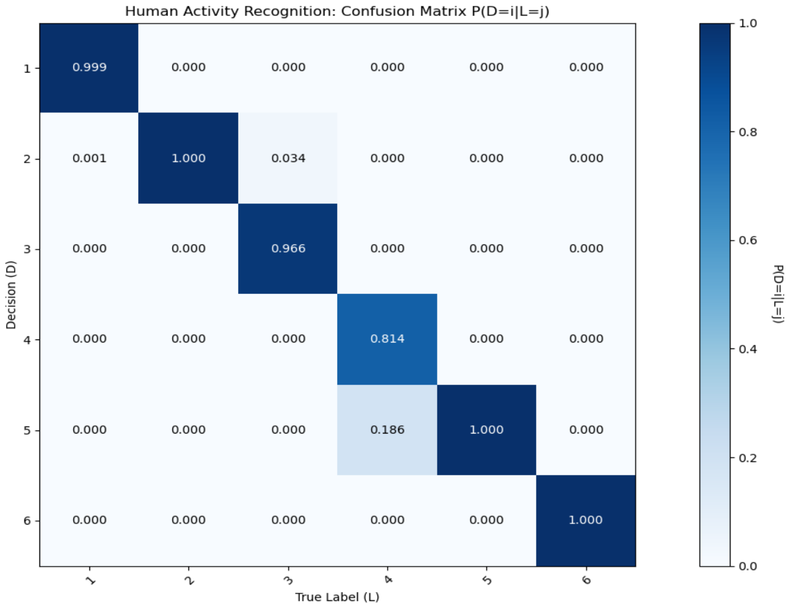
```
=======================================================
DETAILED RESULTS FOR HUMAN ACTIVITY RECOGNITION
=======================================================

Overall Results:
  Accuracy: 0.9631 (96.31%)
  Error Rate (P(error)): 0.0369 (3.69%)
  Total samples: 10299
  Correct: 9919
  Incorrect: 380

Per-Class Results:
Class    N_true     N_correct    Accuracy     Recall
-------------------------------------------------------
1        1722       1721         0.9994       0.9994
2        1544       1544         1.0000       1.0000
3        1406       1358         0.9659       0.9659
4        1777       1446         0.8137       0.8137
5        1906       1906         1.0000       1.0000
6        1944       1944         1.0000       1.0000
```

Confusion Matrix (D=i | L=j) Normalised

Activity	**Walk**	**UpStairs**	**DownStairs**	**Sit**	**Stand**	**Lay**
Walking	0.99	0	0	0	0	0
Walking Upstairs	0.001	1.00	0.034	0	0	0
Walking Downstairs	0	0.10	0.966	0	0	0
Sitting	0	0	0	0.814	0.15	0.02
Standing	0	0	0	0.186	1.000	0.03
Laying	0	0	0	0	0	1.000



For Ex: Walking (N=1,722, 16.7%):

- Recall P(D=1|L=1): ~92%
- Main confusions: Walking Upstairs (4%), Walking Downstairs (4%)
- Excellent performance: Distinct periodic motion pattern clearly identified

KEY OBSERVATIONS:

1) Excellent overall Accuracy (96.31%): The classifier achieves near human-level performance on activity recognition, given the complex, high-dimensional feature space (561 features), The data is also a real one with noise and utilizing a simple gaussian model.
2) The confusion matrices are meaningful and we get a balanced performance across the classes with proper regularization, balanced priors and the laying is most distinctive.

VISUALIZATION:

The Principal Component Analysis (PCA) shows that

A) 1st 20 PCs explicate 90% and above variance
B) Clear clustering in PC space, where Dynamic activities, static activities form different clusters laying is most separated. This strong structure in PC space validates the reason why gaussian model works well.

FINAL CONCLUSION:

Aspect	**Wine Quality (white)**	**HAR**
Samples	14,898	10,299
Features	11	561
Classes	6-7	6
Error Rate	69.33%	3.69%
Gaussian Fit	Poor	Excellent
Class Balance	High Imbalance	Well balanced
Class Separation	Overlapping	Well-separated
Feature Distribution	Skewed, non Gaussian	Gaussian-like
Semantic Structure	Ordinal	Nominal
Regularization Role	Helpful	Essential
Main Confusions	Adjacent qualities	Similar activities

The gaussian MAP classifier achieved:

1) 30.67% accuracy on white wine quality
2) 96.31% accuracy on HAR quality.

CONCLUSION:

From the key observations that we have obtained, we observe that the Gaussian Model is highly appropriate for Human Activity Recognition due to balanced classes, well-separated activities and similar features. While those of Wine Quality, Gaussian is highly inappropriate due to class imbalance, non-gaussian features and significant overlap. Therefore, the appropriateness of a model matters a lot rather than the sophistication.
Proper understanding of the data characteristics and matching them to model assumptions is the correct path for better, accurate results and estimations.

REFERENCES:

Theoretical formulae – Notes and external websites (for understanding)
CODING – Completely based on Python.
Links:

1) ERM & Bayesian - https://am207.github.io/2018fall/wiki/generativemodels.html#erm
https://am207.github.io/2018fall/wiki/generativemodels.html#bayes

2) Fisher LDA - https://github.com/topics/fisher-discriminant-analysis?o=desc&s=