Assignment 1: Introduction to AI/ML: UMC-203

KOLIPAKA BHARGAV SASHANK
SR Number: 23634

07 March 2025

Question 1

Introduction

In this report, we implement the Fisher Linear Discriminant (FLD) for multi-class classification using the CelebA dataset. The goal is to analyze how FLD's estimates change with different training sample sizes, compute FLD weights, determine optimal thresholds for classification, visualize the results, and evaluate classification accuracy.

Part 1: Analysis of FLD Estimates

Estimation is a core component of FLD, as it relies on sample statistics to determine class separation. We analyze how the estimates change with different training sizes n by computing:

- **L2 norms of class mean vectors** to observe changes in mean estimation.
- **Frobenius norms of covariance matrices** to evaluate class variance stability.

We computed these norms for $n = 50, 100, 500, 1000, 2000, 4000$ and visualized the results in Figures 1 and 2.

Part 2: Multi-Class FLD Implementation

Computation of FLD Weights

FLD weights were computed for sample sizes $n = 2500, 3500, 4000, 4500, 5000$ using 20 different subsets (except for $n = 5000$). We solved the eigenvalue problem:

$$S_B W = \lambda S_W W \tag{1}$$

where:

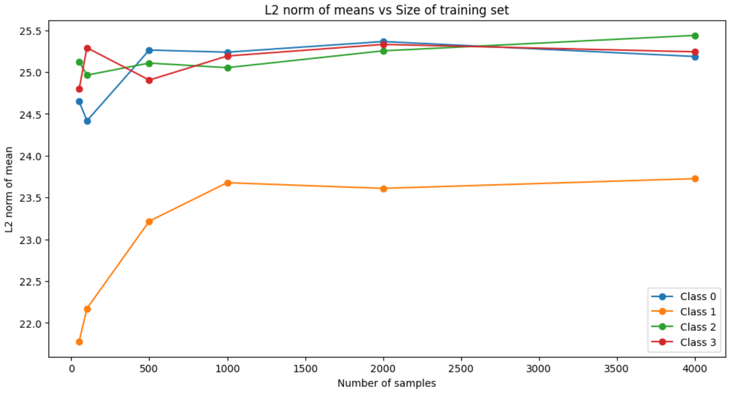

Figure 1: L2 Norms of Mean Vectors for Different Sample Sizes.

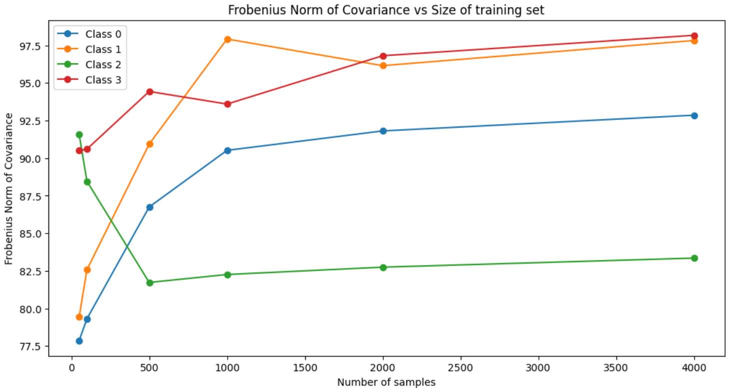

Figure 2: Frobenius Norms of Covariance Matrices for Different Sample Sizes.

- S_W is the within-class scatter matrix.
- S_B is the between-class scatter matrix.
- W contains the top $C-1$ eigenvectors for class separation.

Box Plots of Objective Function

Figure 3 shows the box plot of the multi-class objective function $J(W)$ for different n values.

Scatter Plot of Projected Points

Figure 4 presents the projection of class points in FLD space for $n = 5000$.

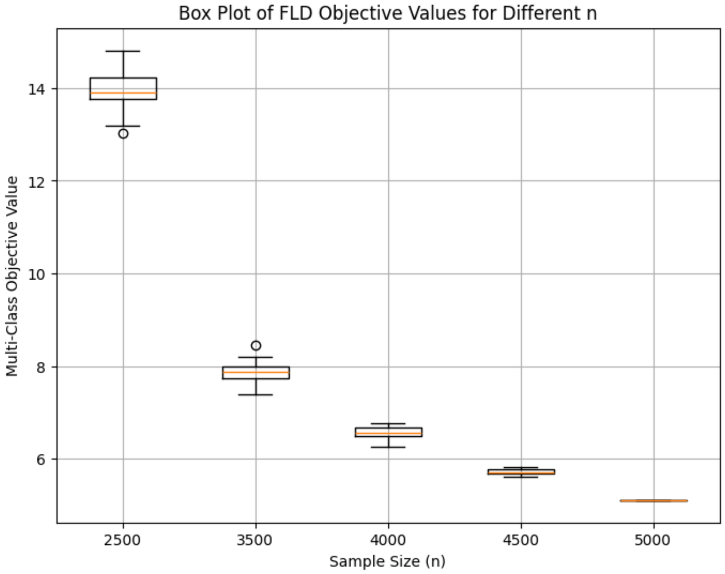

Figure 3: Box plot of multi-class objective values for different sample sizes.

Chosen Thresholds for Classification

Table 1 shows the optimal thresholds chosen based on class means in the projected space.

Discriminant	**Class Transition**	**Threshold**
Disc 1	3 → 1	-0.013419
Disc 1	1 → 0	0.008410
Disc 1	0 → 2	0.031376
Disc 2	0 → 1	-0.006739
Disc 2	1 → 3	0.011328
Disc 2	3 → 2	0.028324
Disc 3	3 → 0	-0.010121
Disc 3	0 → 2	-0.003655
Disc 3	2 → 1	0.003782

Table 1: Optimal thresholds for classification.

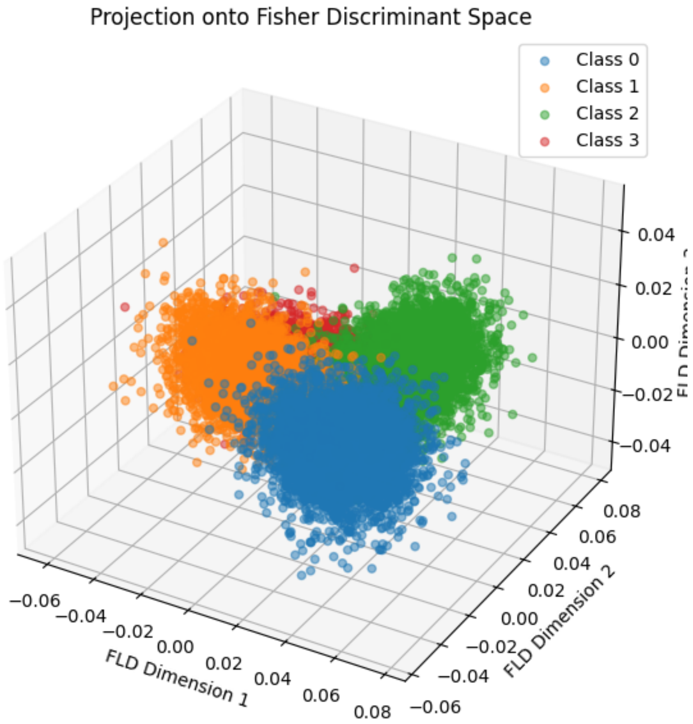

Figure 4: 3D Scatter plot of projected points using FLD (n=5000).

Part 3: Test Set Evaluation

Test Accuracy for Different Sample Sizes

The classifier was evaluated on the test set using the selected thresholds. The accuracy obtained on the test set for the weight matrix computed on the complete training data is 81.9%.

Conclusion

This report successfully implemented multi-class FLD for the CelebA dataset. We computed FLD weights, visualized projections, determined thresholds, and evaluated accuracy. The results suggest that FLD alone does not provide optimal class separation in this dataset, and alternative classification techniques may be needed.

Question 3

0.1 Introduction

The goal of this experiment is to train and evaluate a Decision Tree classifier to predict the presence of heart disease using the UCI Heart Disease dataset. The classifier is trained based on specific hyperparameters obtained from an oracle.

0.2 Dataset and Preprocessing

The dataset consists of 303 instances with 14 features such as age, cholesterol, blood pressure, and chest pain type. The preprocessing steps included:

- Handling missing values by replacing '?' with the mode for categorical variables and the mean for integer variables.
- Splitting the dataset into an 80% training set and a 20% test set.
- Converting the target variable into a binary classification problem: values 2, 3, and 4 were mapped to 1 (Disease).

0.3 Model Training

The Decision Tree classifier was trained using the hyperparameters obtained from the oracle:

- Criterion: `gini`
- Splitter: `random`
- Max Depth: `4`

The classifier was trained using the `DecisionTreeClassifier` from `sklearn`.

0.4 Results and Visualization

0.5 Decision Tree Visualization

The trained decision tree was visualized using the `dtreeviz` module.

0.6 Performance Metrics

The model was evaluated on the test set using the following metrics:

- **Accuracy**: `0.7704918032786885`
- **Precision**: `0.7727272727272727`
- **Recall**: `0.6538461538461539`
- **F1-score**: `0.7083333333333334`

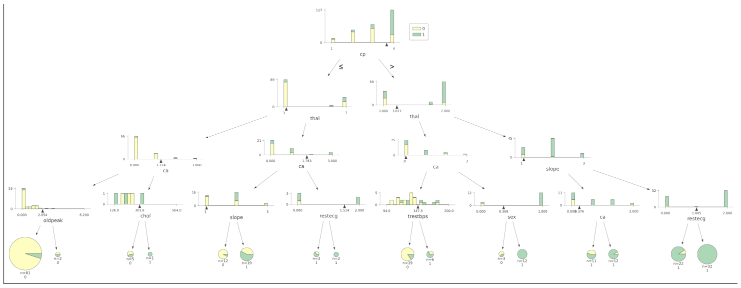

Figure 5: Decision Tree Visualization

0.7 Feature Importance Analysis

The following table ranks the features based on their importance:

Rank	Feature Name	Importance Score
1	cp	0.553676
2	thal	0.162689
3	ca	0.137858
4	slope	0.022893
5	thalach	0.005042
6	oldpeak	0.000000
7	sex	0.048600
8	fbs	0.000000
9	exang	0.053836
10	chol	0.009335
11	restecg	0.006071
12	trestbps	0.000000
13	age	0.000000

Table 2: Feature Importance Ranking

The most important feature for predicting heart disease, according to the trained decision tree, was `cp`.

0.8 Conclusion

A Decision Tree classifier was successfully trained and evaluated for heart disease prediction. The visualization provides insight into how different features influence classification. The most significant feature contributing to heart disease prediction was **cp**.