**BITS F464**

**Machine Learning - Assignment 1**

**FISHER’S LINEAR DISCRIMINANT ANALYSIS**

|  |  |
| --- | --- |
| **Name** | **ID** |
| **Praneetha Vaddamanu** | **2017AAPS0319H** |

**Problem Statement and Methodology**

The given task is to implement Fisher’s Linear Discriminant Analysis for the given datasets, to transform the data to one dimension in such a way so as to separate them optimally. Fisher’s LDA finds the optimal direction *w* along which data, when projected, may be separated. It assumes that the data is linearly separable. The separator here is then the *discriminant point* in one dimension, which decides the class label. The points of the dataset after projection are compared with this discriminant point, to ascertain which side of the point they lie on. Points on one side of the discriminant point are classified as positive points, and those on the other side, as negative points. This discriminant point is determined by solving for the intersection of the normal distribution probability density functions representing the positive and negative data points after projection.

Further, the visualization of projected points, the normal distributions of the projected classes, and the discriminant point is performed for improving clarity and intuition.

**Dataset Description and Preprocessing**

It is required to test the implementation on two datasets, the first being datasets/a1\_d1.csv and the second, datasets/a1\_d2.csv .

Both datasets consist of 1000 data points each. Each row in each dataset corresponds to a single data point, with the last column of each row indicating the class label, which in this case, is either 0 (negative class) or 1 (positive class). The rest of the columns in each row, contain values for the features.

datasets/a1\_d1.csv consists of data points having two features each, while datasets/a1\_d2.csv consists of data points having three features each. The algorithm is evaluated on each of these datasets independently.

**Metrics for Performance Evaluation**

**Accuracy**

The accuracy calculated as,

captures how many data points (as a fraction of the total number of data points fed to the algorithm) were classified correctly. It ranges between 0 and 100 %. A higher accuracy is indicative of better performance by the model.

Relying only on accuracy to judge model performance however, may, in some cases, be misleading. Hence, we also calculate the F-score in order to better evaluate the model’s performance.

**F1-score**

The F1-score is calculated as,

where

and

.

Recall captures how many data points were correctly classified as positive points, as a fraction of the total number of actual positive data points in the dataset.

Precision captures the number of data points correctly classified as positive points, as a fraction of the total number of points classified as positive by the model.

Ideally, we would want a high recall, as well as high precision, but often there is a tradeoff between the two. In order to capture correctly the effect of both recall and precision on the evaluation of performance of the model, we use the *F1-score* which is the harmonic mean of recall and precision.

The F1-score ranges between 0 and 1. A higher F1-score is indicative of better performance by the model.

**Brief Description of Algorithm**

The optimal direction along which the dataset must be projected so that the positive and negative points are separable, is given by the eigenvector (with highest magnitude of eigenvalue) obtained on performing the eigendecomposition of

where is the within-class covariance matrix calculated as

and is the between-class covariance matrix calculated as

where:

denotes class 1, and denotes class 2.

denotes the number of examples in class 1 and denotes the number of examples in class 2.

and are the independent means of classes 1 and 2, respectively.

The projected points are obtained by

where is the optimal direction along which points should be projected, after applying the algorithm,

and is the original feature matrix for the data points.

The discriminant point is calculated by solving for the intersection of the two normal distributions, one obtained for each class after projection.

**Results**

1. **Dataset 1 -** datasets/a1\_d1.csv

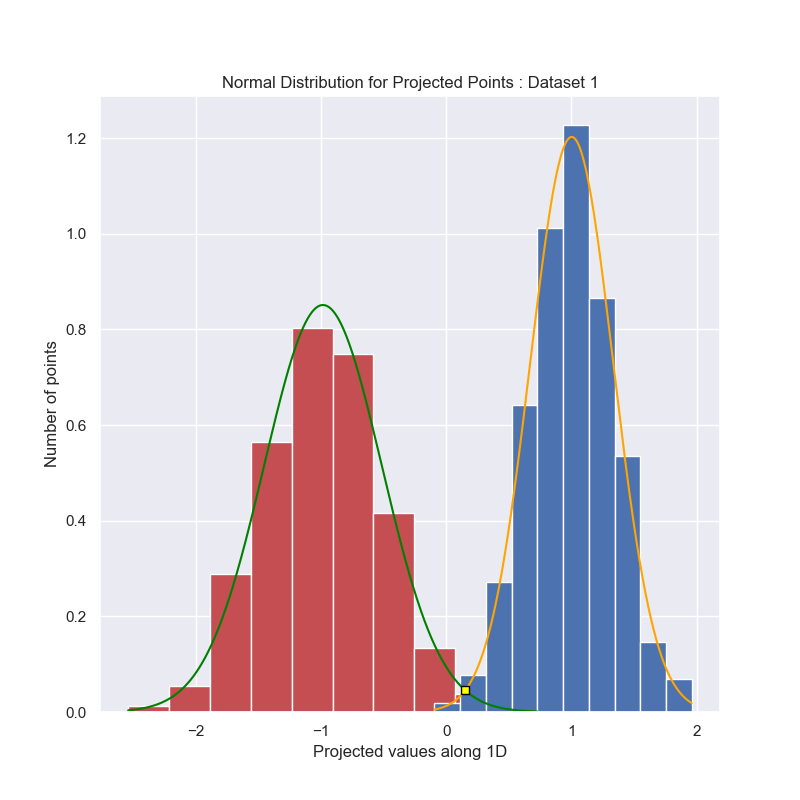
**Accuracy : 99.3 %**

**F1-score : 0.9929**

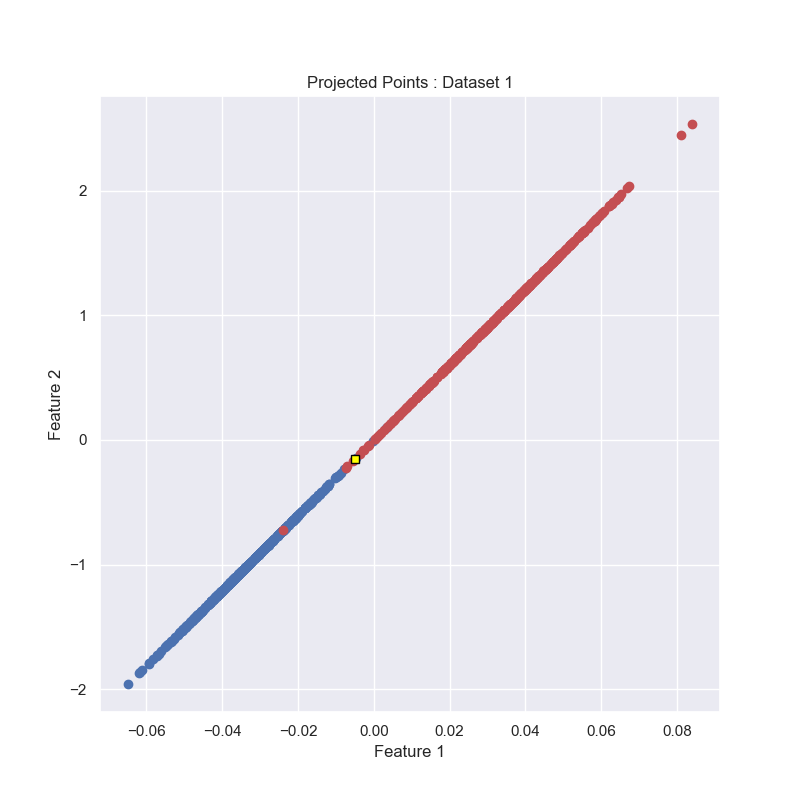
**Visualization:**

The positive class is shown in red and the negative class, in blue. The discriminant point is shown as a yellow square.

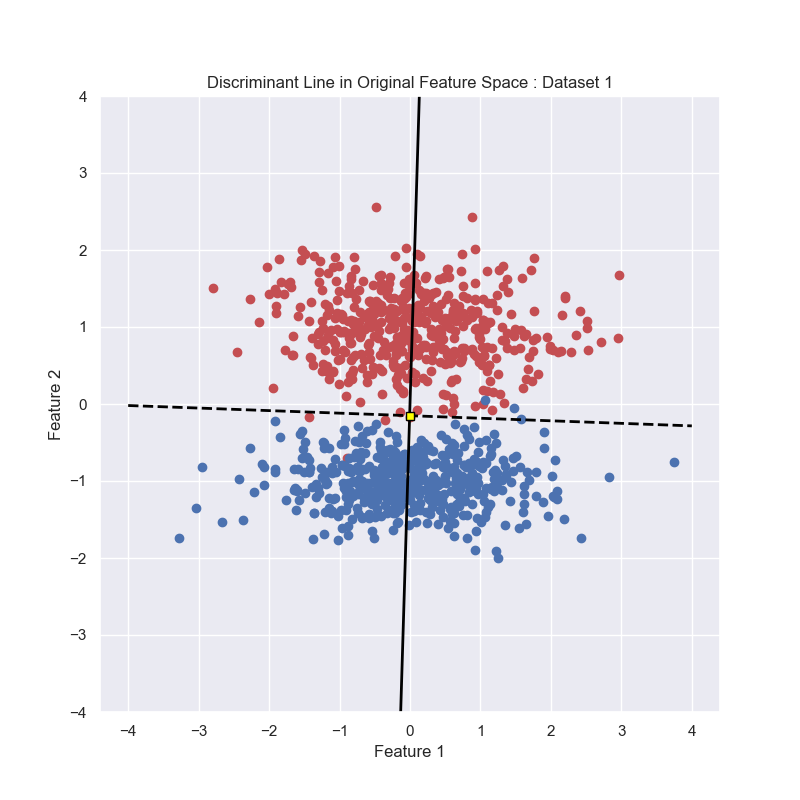
**i) Plotting the normal distribution of the projected points and showing the discriminant point**

****

**ii) Showing the projected points and discriminant point in the original feature space**

****

**iii) Showing the normal line to the discriminant and the discriminant line along with original points in the original feature space**

****

The dashed black line is the discriminant line (normal to the line along with points are projected), whereas the solid black line is the line along which points are projected. The intersection of the two gives the discriminant point, which is shown as a yellow square.

1. **Dataset 2 -** datasets/a1\_d2.csv

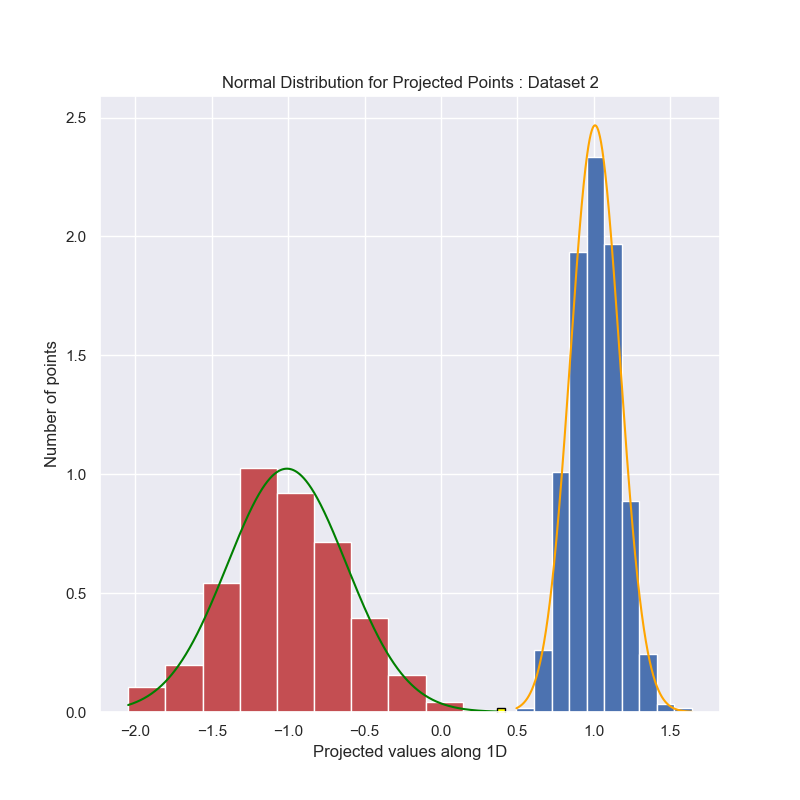
**Accuracy : 100 %**

**F1-score : 1**

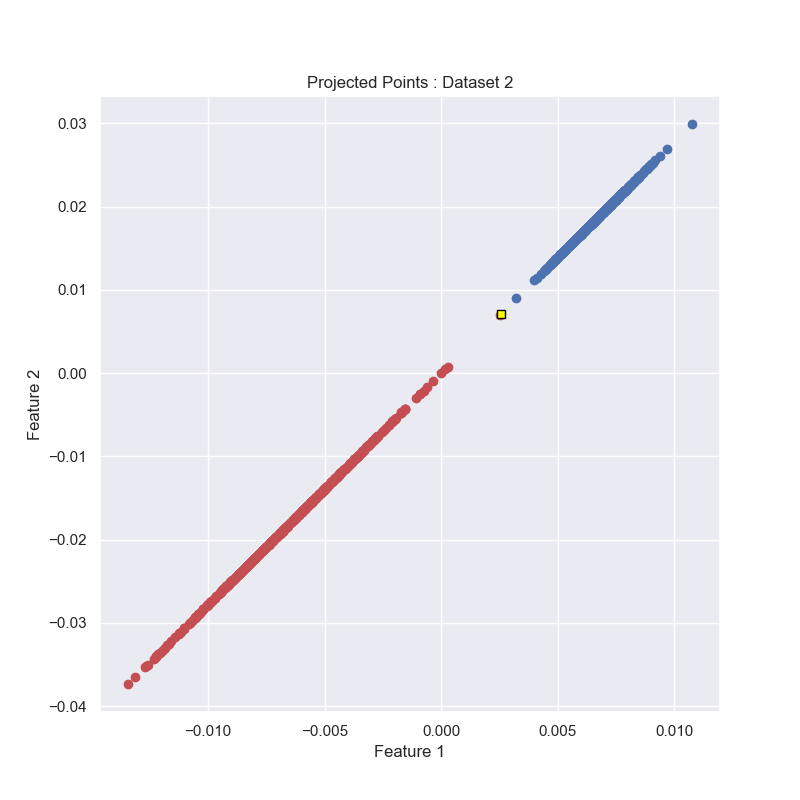
**Visualization:**

The positive class is shown in red and the negative class, in blue. The discriminant point is shown as a yellow square.

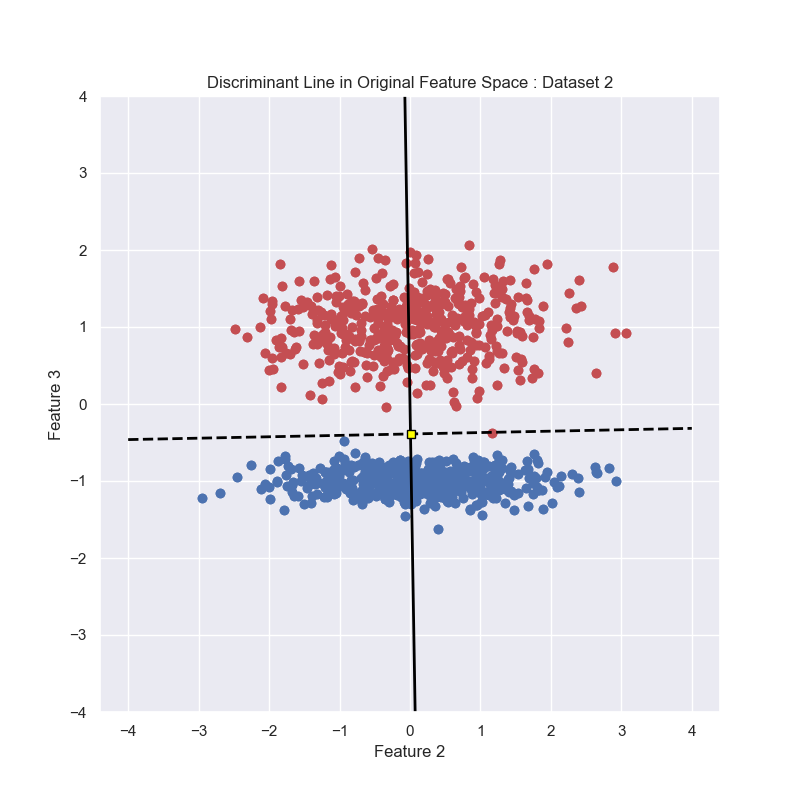
**i) Plotting the normal distribution of the projected points and showing the discriminant point**

****

**ii) Showing the projected points and discriminant point in the original feature space**

****

**iii) Showing the normal line to the discriminant and the discriminant line along with original points in the original feature space, projecting on only the yz plane (i.e. feature space for features 2 and 3), since the separation can be seen clearly in this plane.**

****

The dashed black line is the discriminant line (normal to the line along with points are projected), whereas the solid black line is the line along which points are projected. The intersection of the two gives the discriminant point, which is shown as a yellow square.