**Name: Tingyang Wei**

**Matriculation Number: G2202458H**

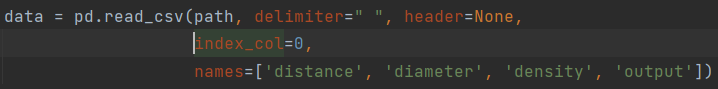
Serial Number: 32

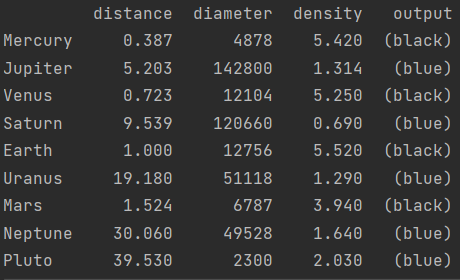
**Problem Description:**

Determine the vector W that optimises the cluster distance and within cluster scatter using FDA. Show all working steps. You may assume that data points labelled black as cluster 1 and blue as cluster 2. Index 1-24, 25-end to use the following data pairs: (d1, d3) and (d2, d3) respectively. Prepare the scatter plot for the input space and draw the projections onto the line defined by W.

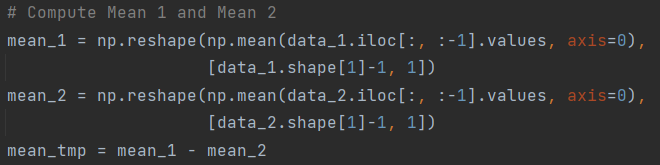
**Solution:**

1. First we put all the data in planet datasets in a txt file and fetch the data using Pandas package in Python.

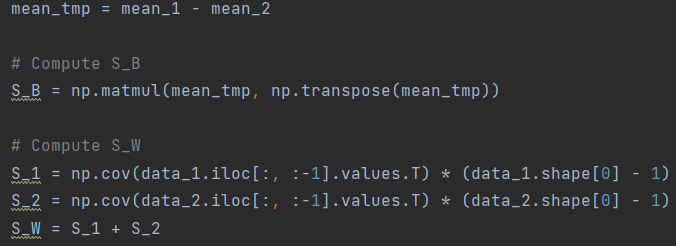




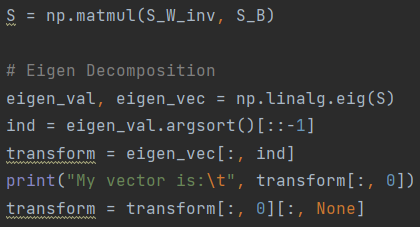
1. To implement the Fisher Linear Discriminant Analysis (FLDA), we should compute all the needed vectors and matrices including ,,,.
2. To compute and, the steps in our script is as follows:



1. Subsequently, we compute the matrices representing the mean difference of two class and the sum of two variance matrices as follows:

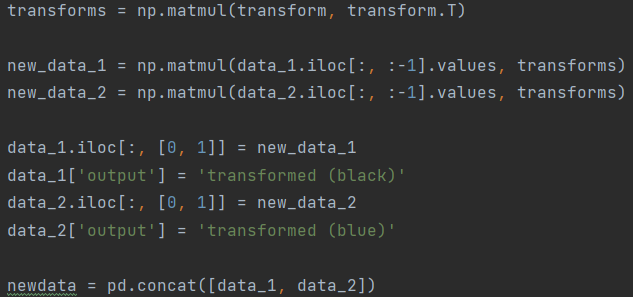


1. Considering the fact , the projection direction we aim to find should be the eigenvector corresponding to the largest eigenvalue of matrix . Therefore, to derive the desirable direction , we need to conduct eigen decomposition on like follows:



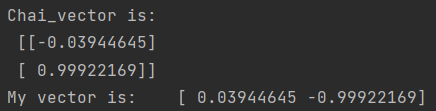
It is worth noting that in the course of CE7429, Prof. Chai derived a vector  using a simple rule as . Later we will show that the derived vector by eigen decomposition is the same as that using the simple rule introduced in CE7429.

1. Having found the eigenvector with the largest eigenvalue, we can normalize it and apply the linear transformation towards the given categorical data distribution as follows.

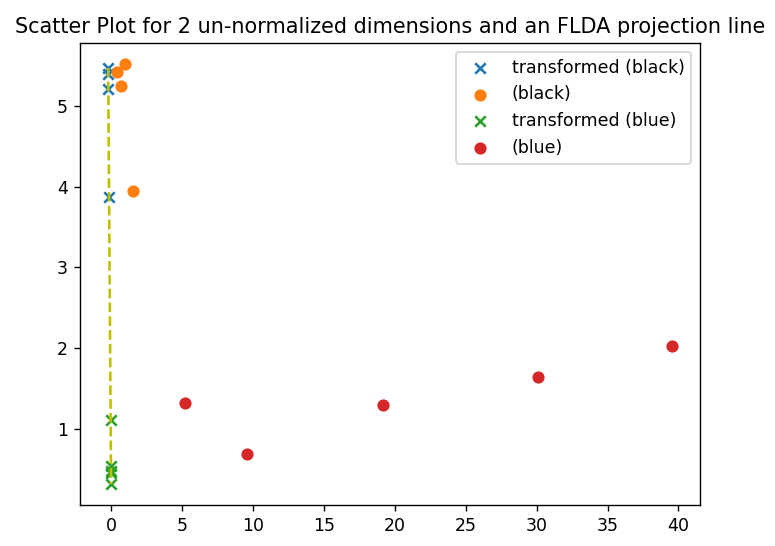


1. Since now CE7429 class have only no more than 20 students I have no idea I should do (d1, d3) or do (d2, d3). Hence in this assignment I do both also do each data in normalized version and un-normalized version of FLDA projection.
2. **For un-normalized (d1, d3)**

The vector  should be as:



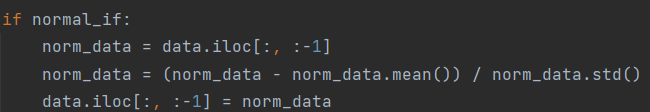
Where herein Chai\_vector represents the vector computed with the rule introduced in the course as . Hence, it has been verified that from a view of projection the direct computation of vector is the same of the eigen decomposition way. And the visualization result is as:

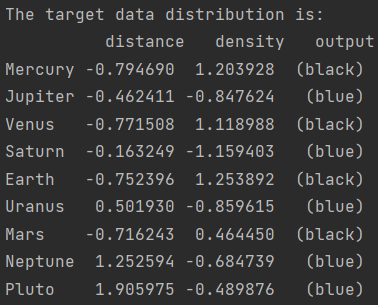


However, since the data has not been normalized the LDA decomposition results seem to be dominated by one direction. So, we consider normalization for better visualization effect for a better understanding.

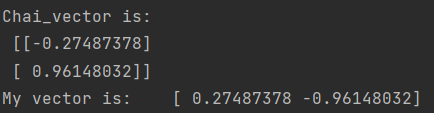
1. **For normalized (d1, d3)**

We generate the normalized data first as follows:

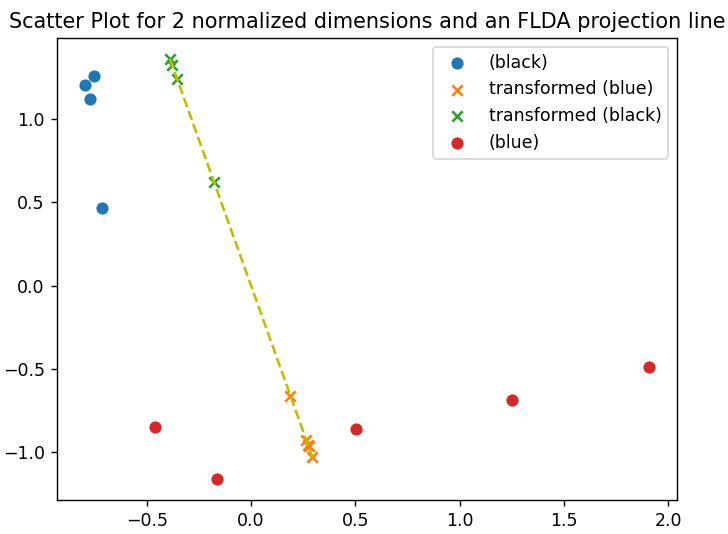




The vector  should be as:

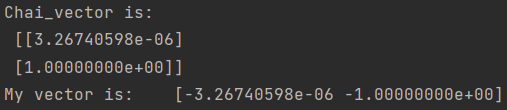


Where herein Chai\_vector represents the vector computed with the rule introduced in the course as . Hence, it has been verified that from a view of projection the direct computation of vector is the same of the eigen decomposition way. And the visualization result is as:

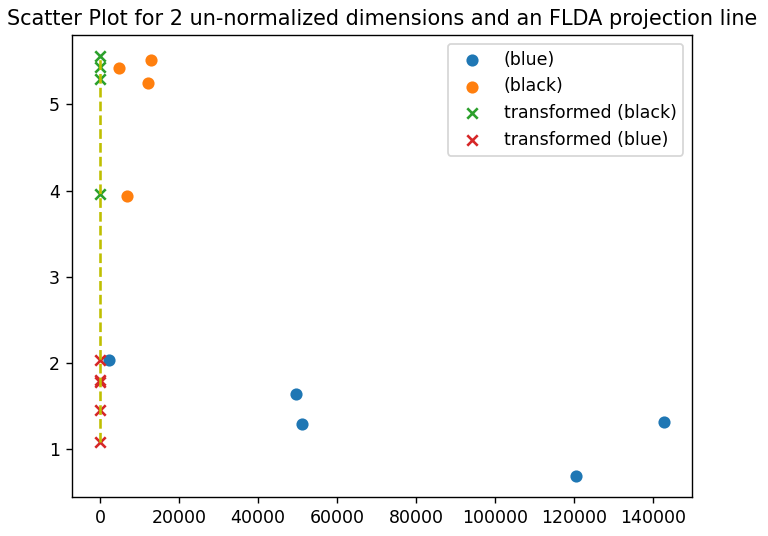


1. **For un-normalized (d2, d3)**

The vector  should be as:



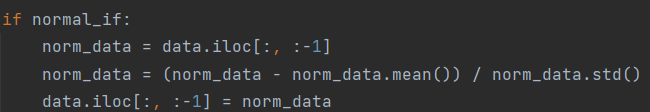
Where herein Chai\_vector represents the vector computed with the rule introduced in the course as . Hence, it has been verified that from a view of projection the direct computation of vector is the same of the eigen decomposition way. And the visualization result is as:

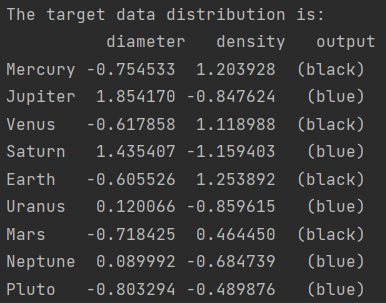


However, since the data has not been normalized the LDA decomposition results seem to be dominated by one direction. So, we consider normalization for better visualization effect for a better understanding.

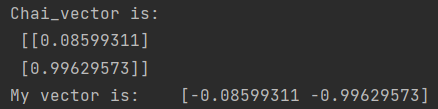
1. **For normalized (d2, d3)**

We generate the normalized data first as follows:





The vector  should be as:



Where herein Chai\_vector represents the vector computed with the rule introduced in the course as . Hence, it has been verified that from a view of projection the direct computation of vector is the same of the eigen decomposition way. And the visualization result is as:

