BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI

Assignment 1

BITS F464 – Machine Learning

Supervised Dimensionality Reduction Techniques

By

Kavya Gupta 2017A7PS0276P

Bhoomi Sawant 2017A7PS0001P

Prachi Agrawal 2018B5A70716P

Submitted to

Dr. Navneet Goyal

Datasets Used

1. Iris Dataset

It is a multivariate dataset with 4 attributes and 150 instances. All the attributes are real-valued.

Attribute Information

1. sepal length in cm

2. sepal width in cm

3. petal length in cm

4. petal width in cm

There are 3 classes, each referring to a type of plant.

a) Iris Setosa

b) Iris Versicolour

c) Iris Virginica

2. Wine dataset

It is also a multivariate dataset with 13 attributes and 178 instances. First attribute named alcohol is class identifier (1-3). All other attributes are continuous.

The attributes are

1) Alcohol

2) Malic acid

3) Ash

4) Alcalinity of ash

5) Magnesium

6) Total phenols

7) Flavanoids

8) Nonflavanoid phenols

9) Proanthocyanins

10) Color intensity

11) Hue

12) OD280/OD315 of diluted wines

13) Proline

Class Distribution: number of instances per class

class 1	59
class 2	71
class 3	48

3. Glass identification dataset

There are 10 attributes (excluding class attribute) and 214 instances. All the attributes are continuously valued.

Attribute Information:

1. Id number: 1 to 214

2. RI: refractive index

3. Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10)

4. Mg: Magnesium

5. Al: Aluminum

6. Si: Silicon

7. K: Potassium

8. Ca: Calcium

9. Ba: Barium

10. Fe: Iron

11. Type of glass: (class attribute)

1) building_windows_float_processed
2) building_windows_non_float_processed
3) vehicle_windows_float_processed
4) vehicle_windows_non_float_processed (none in this database)
5) containers
6) tableware
7) headlamps

4. Handwritten digits dataset

There are 64 attributes. The dataset comprises of 8*8 pixels images. The attributes consist of integer pixel values in the range 0 to 16. The class value is the digit - 0 to 9.

Curse of Dimensionality

Dimensionality refers to the number of features in the dataset. When the number of features is very large relative to the number of observations in the dataset, some algorithms find it difficult to train models effectively. Very high dimension causes all the data points in the dataset to appear equidistant from each other. The distance loses its meaning, thus causing problems. The more features we have, more samples we will need so that all combinations of feature values are well represented by the data. Also, if there are more features than observations then there is a risk of overfitting the model which is undesirable. Dimensionality reduction techniques provide a solution to the curse of dimensionality.

Dimensionality reduction

The aim of dimensionality reduction technique is to reduce the number of dimensions either by discarding some attributes and keeping only the relevant ones or by creating a new set of dimensions, but without losing much information. Some advantages of dimensionality reduction techniques are:

1) Space requirements for data storage decreases.
2) Less computation and training time.
3) Removal of redundant and undesired features
4) Visualization of data becomes easier.

There are two major techniques for reducing dimensionality- feature selection and feature extraction.

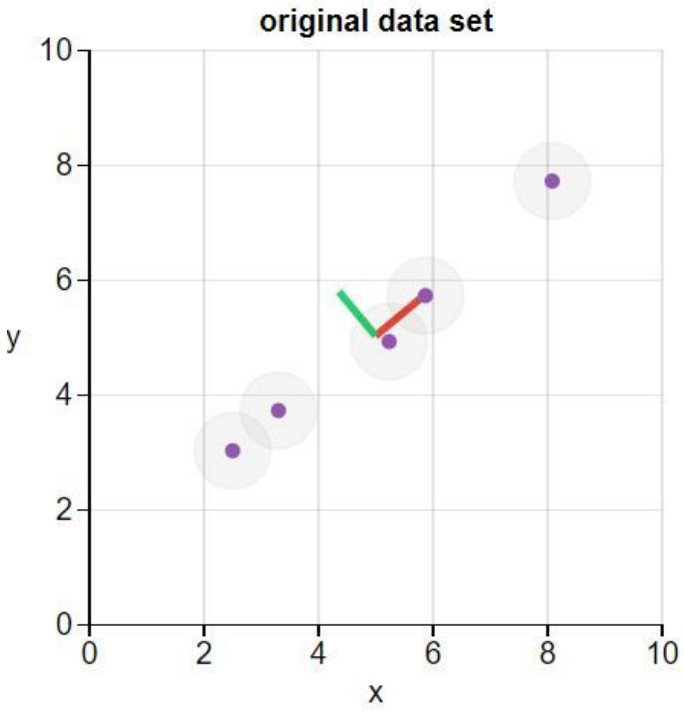
1) Feature Selection: this method involves filtering irrelevant or redundant features from the dataset. The key difference between feature selection and extraction is that feature selection keeps a subset of the original features while feature extraction creates new features.
2) Feature Extraction: this method involves creating a new set of features that still captures most of the useful information. Each new variable is formed by a combination of the input variables.

Dimensionality reduction can be supervised (PCA, SVD) or unsupervised (FLD, Metric Learning). The basic principles, applications and advantages of these dimensionality techniques are explained below. These techniques were applied on different datasets. The corresponding results, comparison and inferences are also discussed in detail.

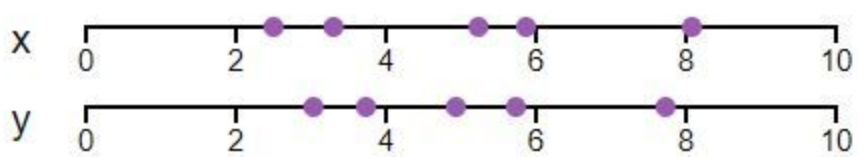
PCA (Principal Component Analysis)

PCA is a technique of feature extraction which combines input variables in a specific way such that the least important variables are dropped whereas the most informative portions of all the variables are retained. Each of the new extracted variables after PCA are independent of one another. This provides an advantage as the linear model assumes that variables are independent of one another. PCA reduces the dimensionality of the dataset while preserving maximum variability as possible. This is achieved by finding new variables that are linear functions of the variables in the original dataset, that successively maximize variance and that are uncorrelated to each other.

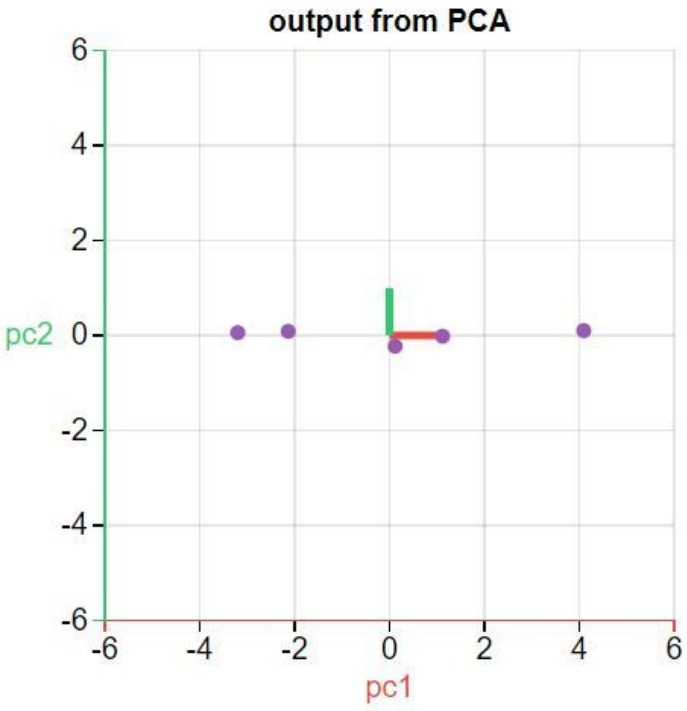
For example consider the following distribution with two dimensions,



Projecting along x and y directions,

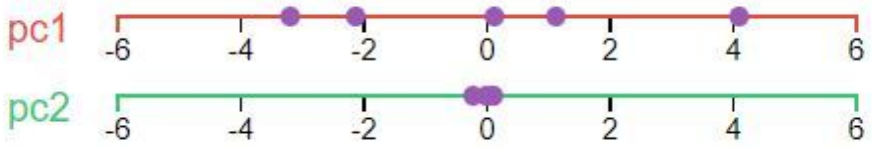


After applying PCA, the new principal components are as shown below,



As can be seen from the distribution above, the principal components are orthogonal to each other. Because the principal components are orthogonal to each other, they are statistically linearly independent of one another.

Projecting along pc1 and pc2,



If the data is to be converted in one dimension, then it is best to choose the direction of the principal component with the most variation. Even if we drop PC2, it won't affect much since it contributes the least to the variation in the data set.

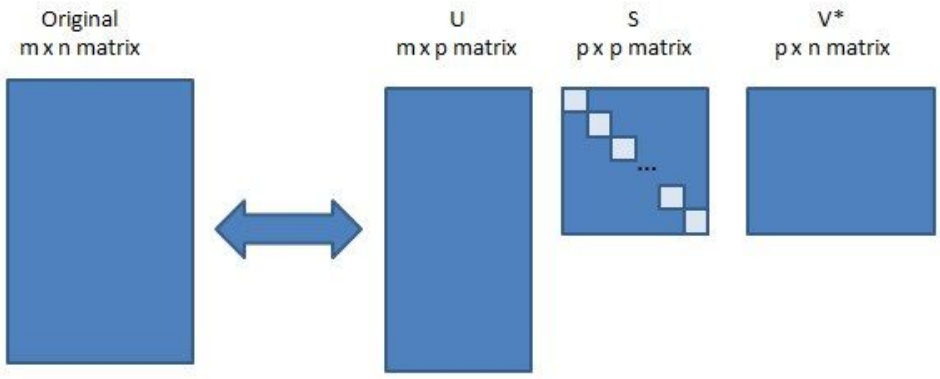
Procedure of applying PCA is as follows:
Firstly, a covariance matrix is constructed which captures the estimates of how every variable relates to every other variable of the distribution. Understanding how one variable is associated with another provides an indication upon their informative capabilities. Then, eigenvalues and eigenvectors are constructed for the covariance matrix. Eigenvectors represent the directions whereas eigenvalues represent the

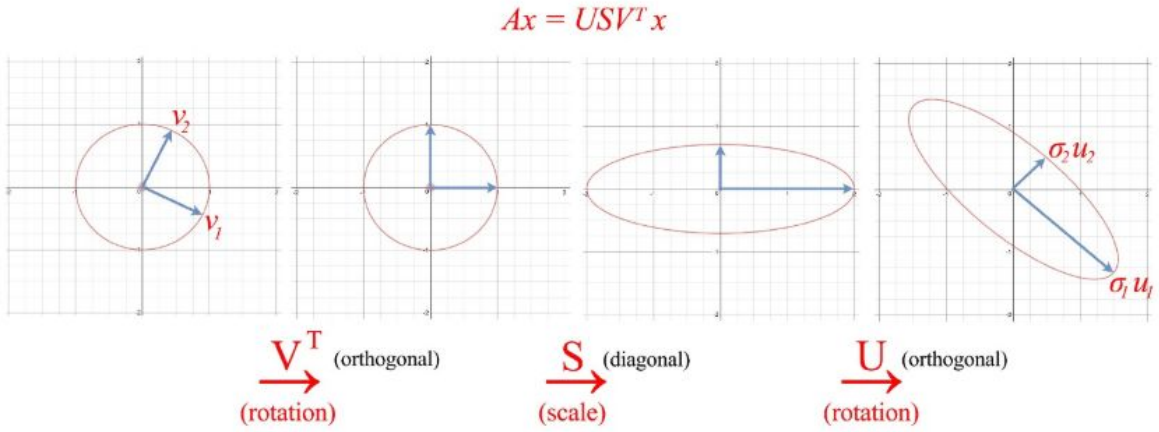
magnitude of relative importance of the different directions. Bigger eigenvalues correspond with directions that are more important. More variability in a particular direction correlates with explaining the behavior of the dependent variable. More variability usually indicates signal, and on the other hand little variability usually indicates noise. Thus, we wish to keep those directions that have more variability. Hence, the eigenvectors are sorted according to their eigenvalues in decreasing order and first x eigen-vectors are chosen which form the new x dimensions.
The original n dimensional data points are represented in x-dimensional vectors and the subsequent analysis is done using the transformed dataset.

In short, PCA finds a new set of dimensions (or directions) such that all the dimensions are orthogonal, thus linearly independent and ranked according to the variance. It means the more important principle axis occurs first. (more important = more variance).

SVD (Singular Value Decomposition)

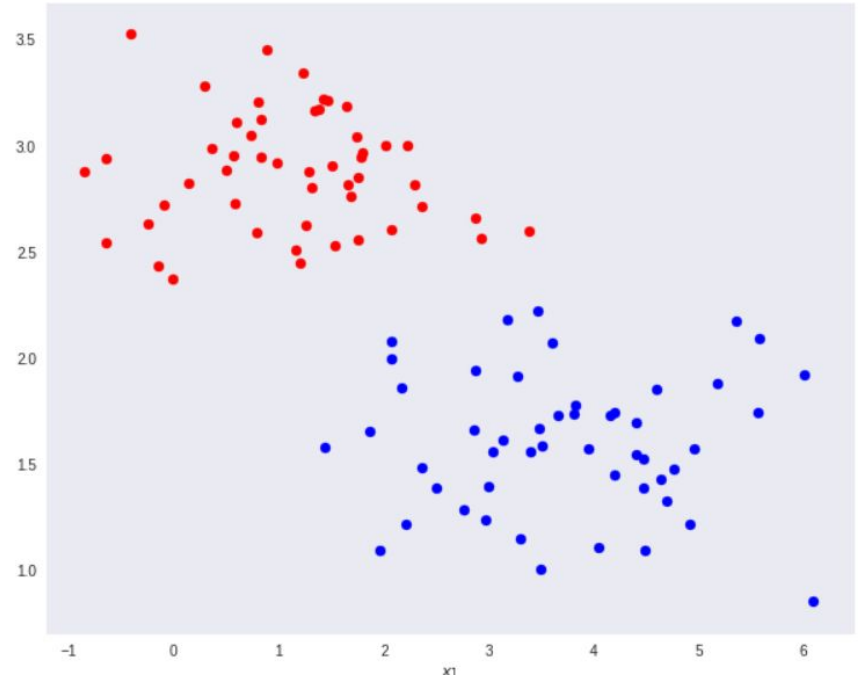
SVD is an algorithm that factors an matrix M of real or complex values into three component matrices in the form USV*. Let M be a matrix of size m x n. Then, U is a m x p matrix. S is a p x p diagonal matrix. V is an n x p matrix, where V* is the transpose of V, a p x n matrix (conjugate transpose in case of M containing complex values). The value p is called the rank. The diagonal entries of S are called the singular values of M. The columns of U are referred to as the left-singular vectors of M and the columns of V form right-singular vectors of M. Depending on the amount of variance we wish to capture, we can choose the number of vectors. One of the features of SVD is that given the decomposition of M into U, S, and V, an approximation of the original matrix M can be reconstructed. The singular values in the diagonal matrix S can be used to understand the amount of variance explained by each of the singular vectors.



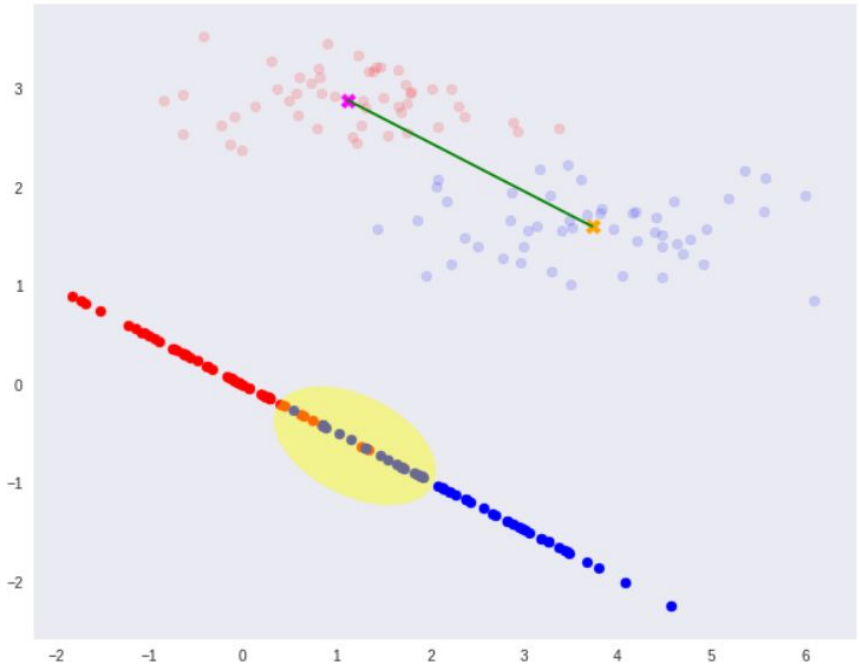


FLD (Fisher's Linear Discriminant)

FLD uses the principle of maximizing the between class variance and minimizing the within class variance.
Consider a 2 class classification problem. The two classes are represented using red and blue.



After projecting using the line joining class means as shown below,

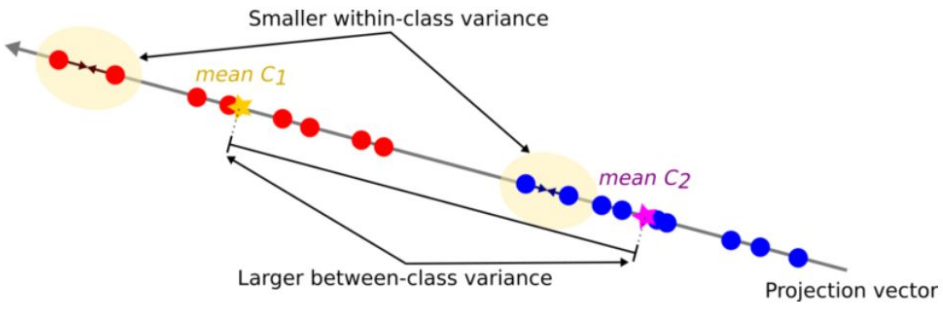


The illustration above shows that there is some class overlapping which is highlighted by the yellow ellipse on the plot. The idea proposed by FLD is to maximize a function that will give a large separation between the projected class means while also giving a small variance within each class, thereby minimizing the class overlap. FLD selects a projection that maximizes the class separation. To do that, it maximizes the ratio between the between-class variance to the within-class variance.

FLD hence maintains the following two properties:

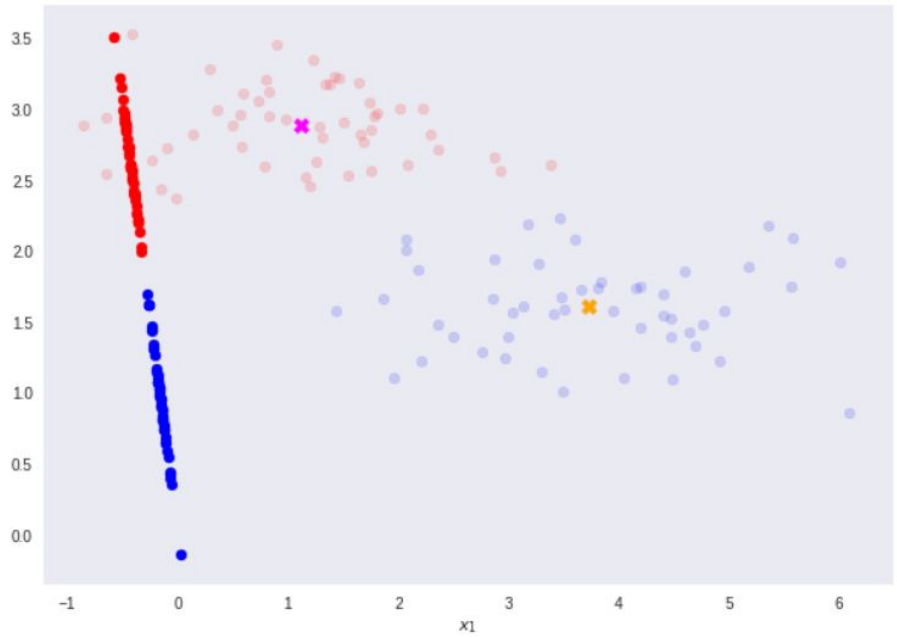
- A large variance among the classes.
- A small variance within each of the classes.

A large between class variance refers that the projected class averages should be as far apart as possible. On the other hand, a small within class variance means keeping the projected data points closer to each other.



To find the projection with the following properties, FLD learns a weight vector W which is the square of between class variance divided by the sum of squares of within class variance.This measure is effectively a measure of the signal to noise ratio for the class labelling. W is directly proportional to the inverse of the within class covariance matrix times the difference of the class means.

As shown below, the classification can now be done simply by choosing a threshold value:



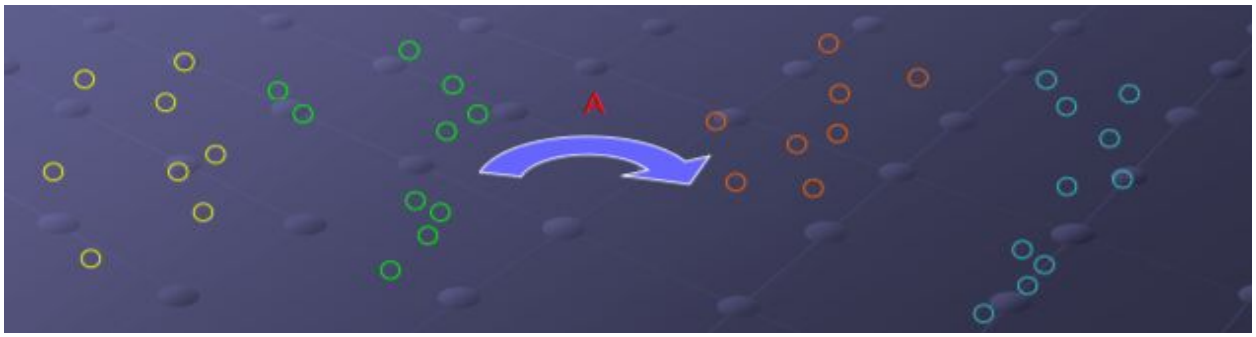
FLD can be generalized for multiple classes by generalizing within class and between class covariance matrices. The maximization of the FLD criterion is solved by an eigen decomposition of the matrix multiplication. Fisher's Linear Discriminant is a technique for dimensionality reduction and not a discriminant. For binary classification, an optimal threshold can be chosen which will classify the data accordingly. For multiclass data, a class conditional distribution can be modeled using a Gaussian distribution.

To find the optimal direction to project the input data, Fisher needs supervised data and hence it is a supervised dimensionality reduction technique.

Metric Learning (NCA- Neighborhood Component Analysis)

Distance metric learning creates task dependent distance metrics from supervised data automatically. This concept can be used to perform various tasks like dimensionality reduction.
Supervised metric learning algorithms take input points and class labels and learn a distance matrix that makes points from the same class (in case of classification) or with close target value (in case of regression) close to each other and points from different classes or with distant target values far away from each other.

NCA is a distance metric learning algorithm that aims to improve the accuracy of nearest neighbors classification. The algorithm focuses on maximizing a stochastic variant of the leave one out KNN score on the training dataset. It can also learn a lower dimension transformation of data that helps in better data visualization and fast classification. Despite the simplicity of objective function used in NCA, its performance is the same as or better than that of other Mahalanobis distance measures for KNN.
NCA finds the best projection without making the assumption regarding any parametric structure in the lower dimension representation. This involves maximizing the cost function f(A) using gradient ascent over a nonsquare A. NCA maximizes the objective function using a gradient based optimizer like deltabar-delta and conjugate gradients.

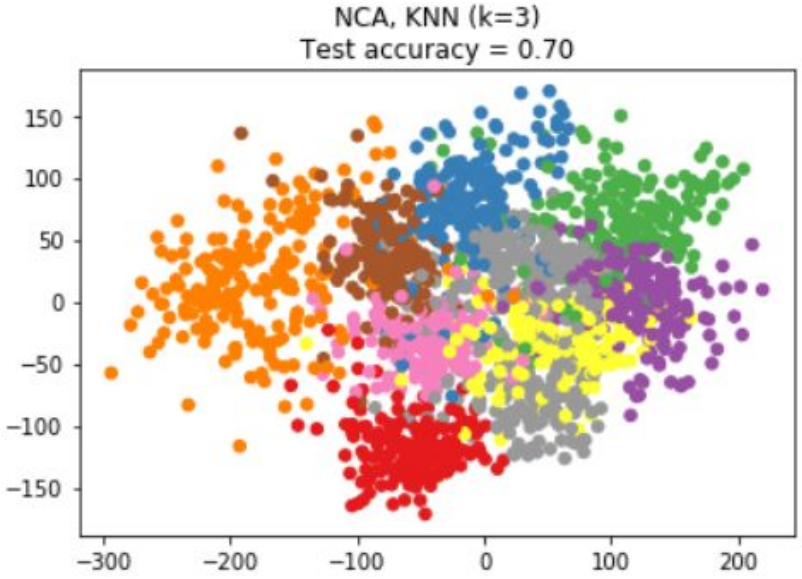


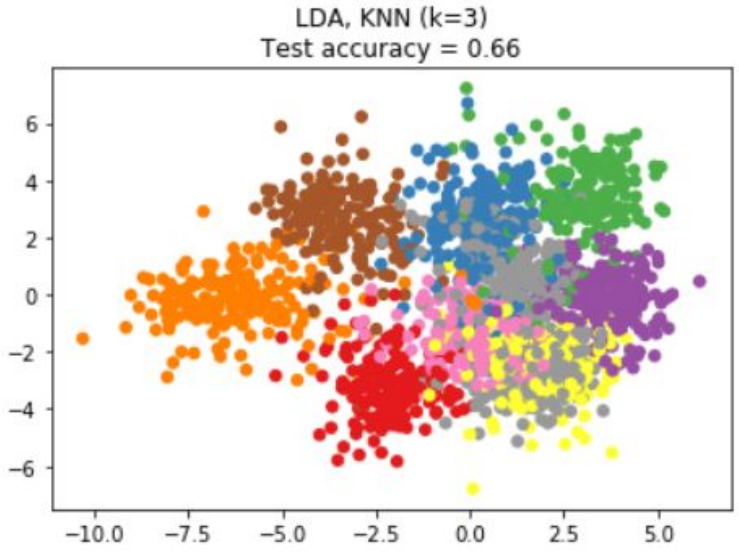
When a two dimensional projection is considered, the classes are much better separated by the NCA transformation than by either PCA (unsupervised) or LDA as shown below:

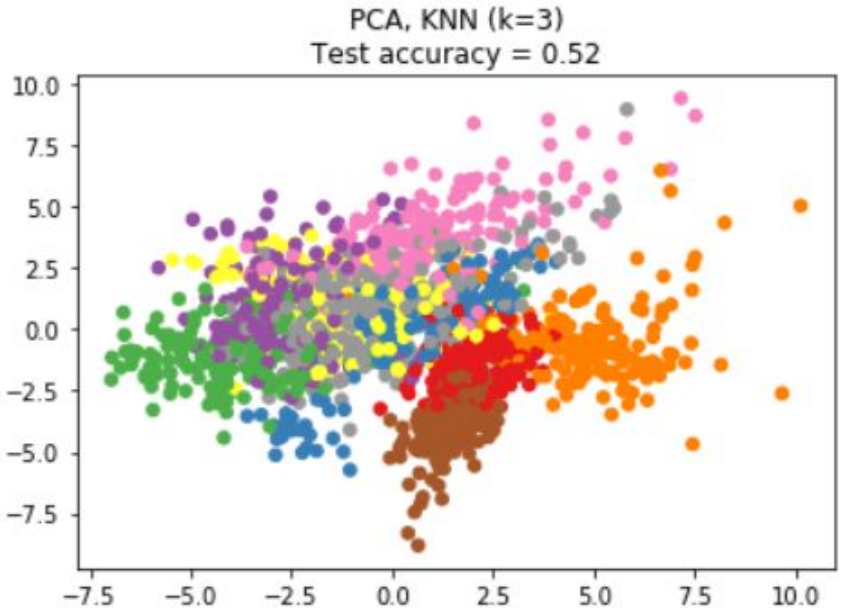
Comparing PCA, LDA and NCA

PCA is an unsupervised dimensionality reduction technique while LDA and NCA (metric learning) are supervised.

Supervised dimensionality reduction aims at finding a lower dimensional representation that maximizes the separation of labeled data while unsupervised dimensionality reduction makes the assumption that the data lies on an embedded lower dimensional manifold within the higher dimensional space. The latter aims at capturing few properties of the original data like the variance or local distance measurements in the low dimensional representation.



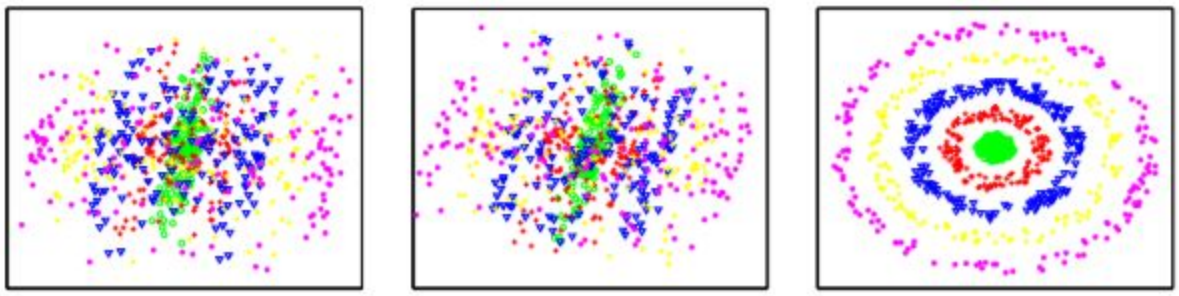




We observed the Dataset visualizations results of PCA, LDA and NCA applied to concentric rings dataset which was performed by a group of researchers.

Source: Neighbourhood Components Analysis (research paper)
Jacob Goldberger, Sam Roweis, Geoff Hinton, Ruslan Salakhutdinov Department of Computer Science, University of Toronto {jacob,roweis,hinton,rsalakhu}@cs.toronto.edu

Concentric rings:



PCA LDA NCA

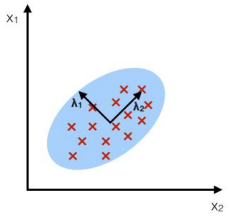
The above results are obtained due to the following reasons:

- If the noise variance is very large, the projection calculated by PCA is forced to include that noise.
- PCA optimizes the loss of variance, it tries to preserve as much variance as possible.
- PCA is convex and has an analytical solution.
- PCA does not guarantee good performance on tasks like classification because being an unsupervised technique, it does not use class labels as input. Also, it makes some assumptions on the distribution of the input data.

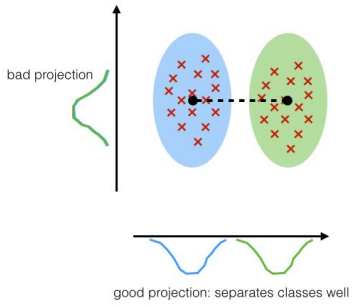
- LDA will not give appropriate results in cases where the classes are not convex and cannot be linearly separated.
- It gives optimal results if all classes have Gaussian distribution with a single shared covariance which might not be the case always
- Doesn't work well when the training set is too small and dimension is high.
- It suffers from a small sample size problem when applied on high dimensional data when the within class scatter matrix is nearly singular.

- NCA adaptively finds the best projection without making any assumption of parametric structure in the lower dimensional representation.
- It makes no assumptions about the class distributions.
- NCA optimizes the overall classification accuracy of the preceding nearest neighbor classifier and tries to preserve the local structure.
- NCA is a non convex optimization problem. It means that every time NCA is used, we may get a different solution and either average or the best solution is taken.

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation



Results

In this section, we discuss the results obtained on performing different dimensionality reduction techniques. We have performed 2 supervised dimensionality reduction techniques - LDA & NCA and 2 unsupervised dimensionality reduction techniques - PCA and SVD.

I. Principal Component Analysis Results

Principal Component Analysis was performed on Wine Dataset.
The princomp function was used in R to perform dimensionality reduction using PCA.

Importance of components is given below-

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5
Standard deviation	2.2014546	1.5244375	1.1790004	1.0001334	0.95375040
Proportion of Variance	0.3728002	0.1787623	0.1069263	0.0769436	0.06997229
Cumulative Proportion	0.3728002	0.5515624	0.6584887	0.7354324	0.80540465
	Comp.6	**Comp.7**	**Comp.8**	**Comp.9**	**Comp.10**
Standard deviation	0.78348294	0.74313782	0.60414255	0.5512616	0.49332440
Proportion of Variance	0.04721889	0.04248106	0.02807602	0.0233761	0.01872069
Cumulative Proportion	0.85262353	0.89510460	0.92318061	0.9465567	0.96527740
	Comp.11	**Comp.12**	**Comp.13**		
Standard deviation	0.43726853	0.40598115	0.308819182		
Proportion of Variance	0.01470798	0.01267851	0.007336099		
Cumulative Proportion	0.97998539	0.99266390	1.000000000		

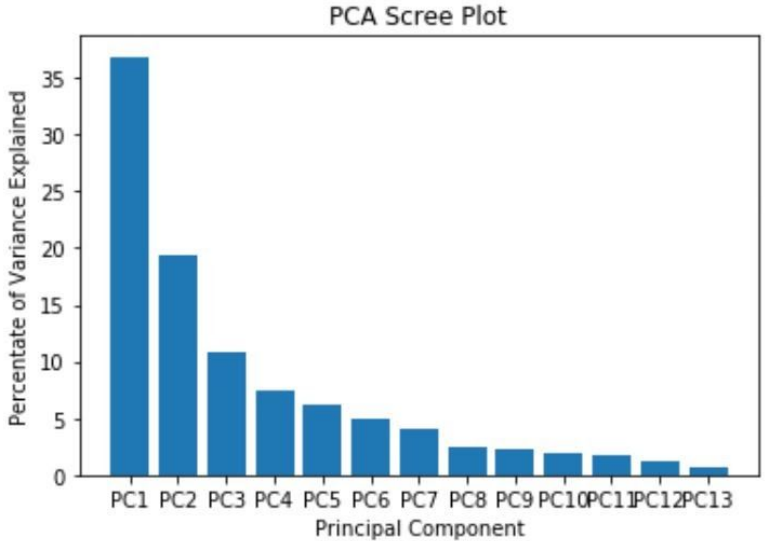


Fig 1.1 PCA Scree plot

Fig 1.1 shows a scree plot. It shows the percentage of variance for each principal component and helps in understanding the number of components we must keep in PCA. Scree plots always display a downward curve like shown above. The point where the slope of the curve is clearly leveling off indicates the number of components that we should keep.

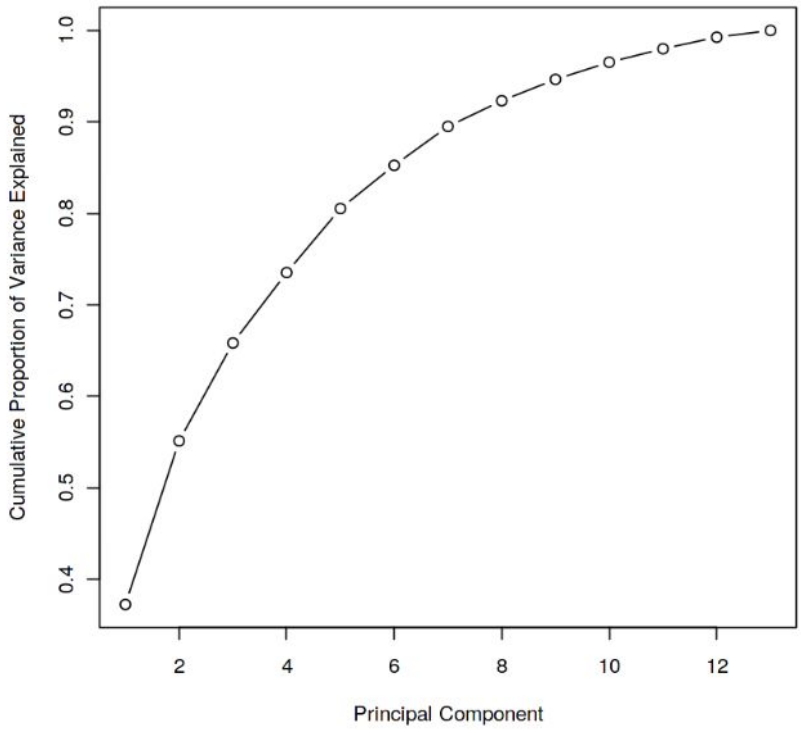


Fig 1.2 Cumulative proportion of variance.

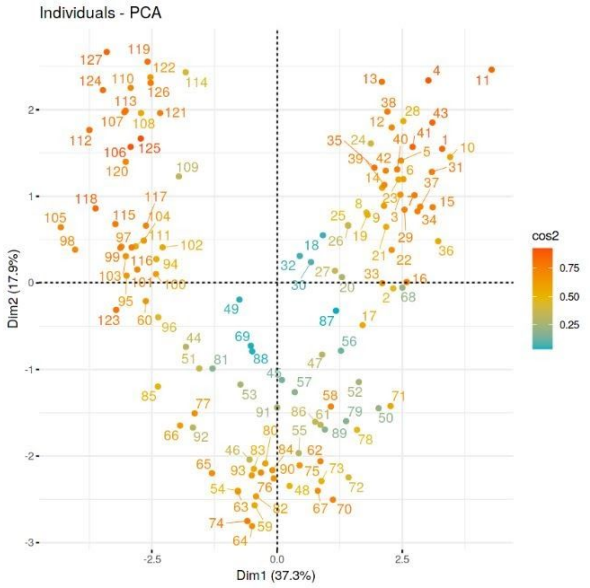
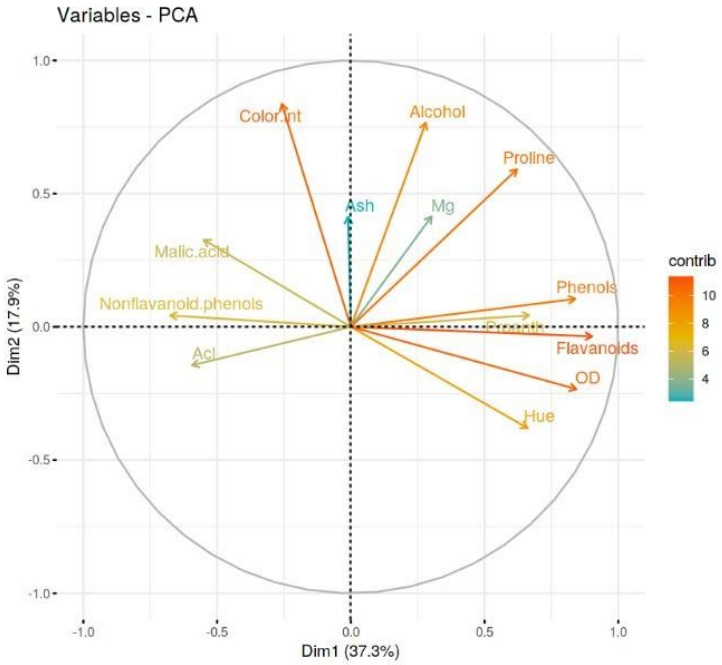
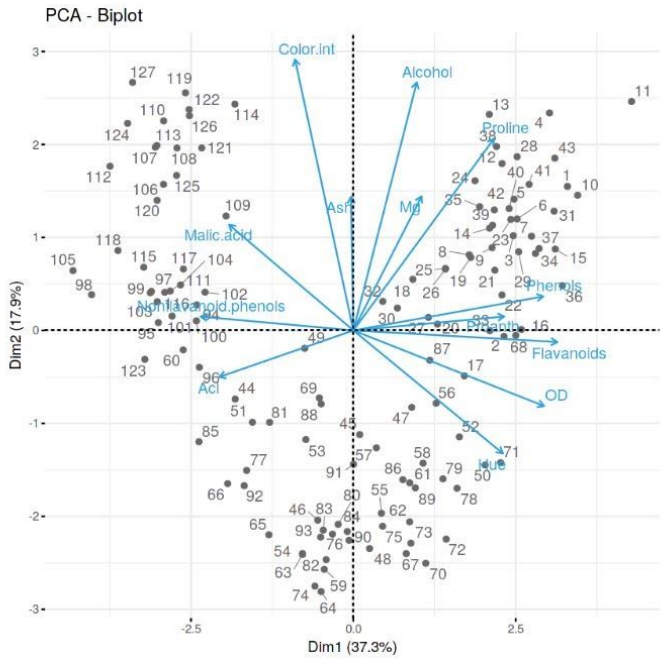


Fig 1.3 The above plot shows a graph of individual data points. Individuals with similar profiles are grouped together.

From the above data and plots, it is observed that the first 7 components contribute ~90% of the information required for the entire data. Hence, 13 components can be reduced to 7 for further analysis with 90% information.



The above plot shows a graph of variables. Positive correlated variables point to the same side of the plot. Negative correlated variables point to opposite sides of the graph.



The above biplot represents both the observations and variables of a matrix of multivariate data on the same plot.
Each attribute on which PCA was applied in the form of an arrow. The arrows for each variable point towards increasing value of that attribute.

Classification Results

We performed classification on the dimensionally reduced data using two classification techniques namely, logistic regression and decision tree classifier. We also compared the accuracy of classification of the given data with dimensionally reduced data.

I. Using Decision Tree Classifier

a. Classification using decision tree without applying PCA

The confusion matrix is shown below-

Class	1	2	3
1	15	1	0
2	1	19	1
3	0	4	10

Accuracy=86.27%

b. For 7 components- Decision tree classifier

Based on the scree plots in Fig. 1.1, it was observed that 7 principal components contribute to approximately 90% of the information. Hence, we reduce the data with 13 attributes to 7 principal components and classify using decision tree.

The confusion matrix is shown below-

Class	1	2	3
1	14	2	0
2	0	21	0
3	0	0	14

Accuracy=96.07%

For 2 components- decision tree classifier

We also checked the accuracy of classification with 2 principal components.

The confusion matrix obtained is the same as that obtained when 7 components were considered.

Class	1	2	3
1	14	2	0
2	0	21	0
3	0	0	14

Accuracy=96.07%

Thus, reducing the dimensionality of wine dataset to 2 dimensions works well for decision tree classifier.

II. Using Logistic Regression

a. Classification using logistic regression without applying PCA

Class	1	2	3
1	16	0	0
2	1	20	0
3	0	0	8

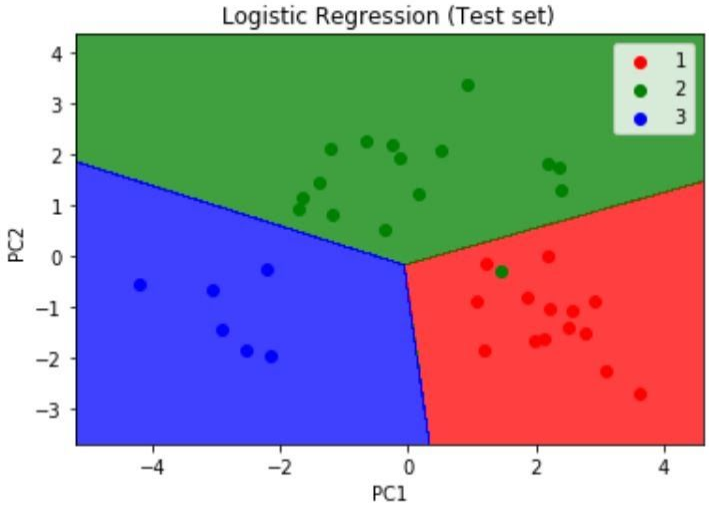
Accuracy=97.8%

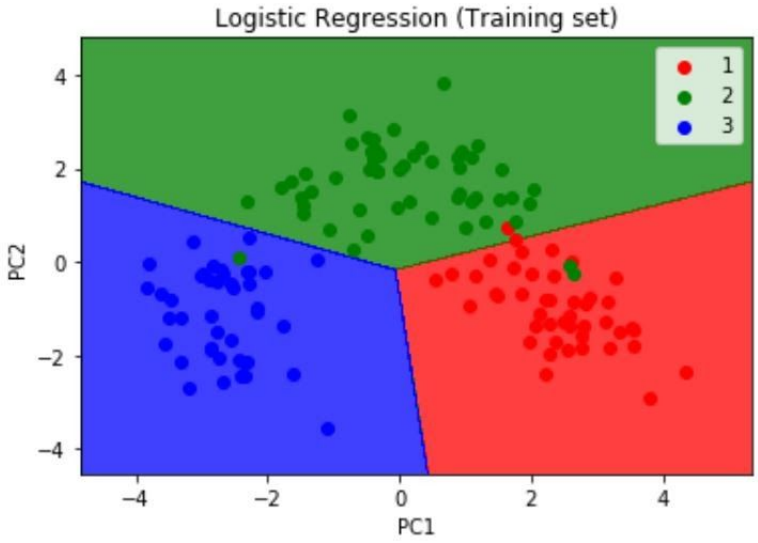
b. For 2 components- Logistic Regression

Logistic regression was applied on the dataset with 2 principal components in order to confirm the sufficiency of 2 components for dimensionality reduction. The confusion matrix is given below-

Class	1	2	3
1	16	0	0
2	1	20	0
3	0	0	8

Accuracy=97.8%





The above graphs show the classification performed by logistic regression on dimensionally reduced data with 2 principal components.

On comparing the classification accuracy with and without application of PCA, it was observed that when classified using decision tree, dimensionally reduced data gave a better accuracy. When logistic regression was used, the accuracy was the same.

From the scree plots, we concluded that the dimensionality of dataset could be reduced to 7 components. However, reducing to 2 components does not compromise with the accuracy of classification (in comparison with the accuracy obtained using 7 components).

II. Singular Value Decomposition(SVD) Results

SVD was performed on the iris dataset. Decision tree classifier was used to classify.

Classification using rpart:

Matrix for the original data

	Setosa	Versicolor	Verginica
Setosa	50	0	0
Versicolor	0	49	5
Verginica	0	1	45

Accuracy=97%

Matrix for reduced data

	Setosa	Versicolor	Verginica
Setosa	50	0	0
Versicolor	0	47	0
Verginica	0	3	50

Accuracy=98.5%

Classification using Decision Tree:

Matrix for the original data

	Setosa	Versicolor	Verginica
Setosa	50	0	0
Versicolor	0	47	1
Verginica	0	4	48

Accuracy=97.5%

Matrix for reduced data

	Setosa	Versicolor	Verginica
Setosa	50	0	0
Versicolor	0	47	0
Verginica	0	3	50

Accuracy = 98.5%

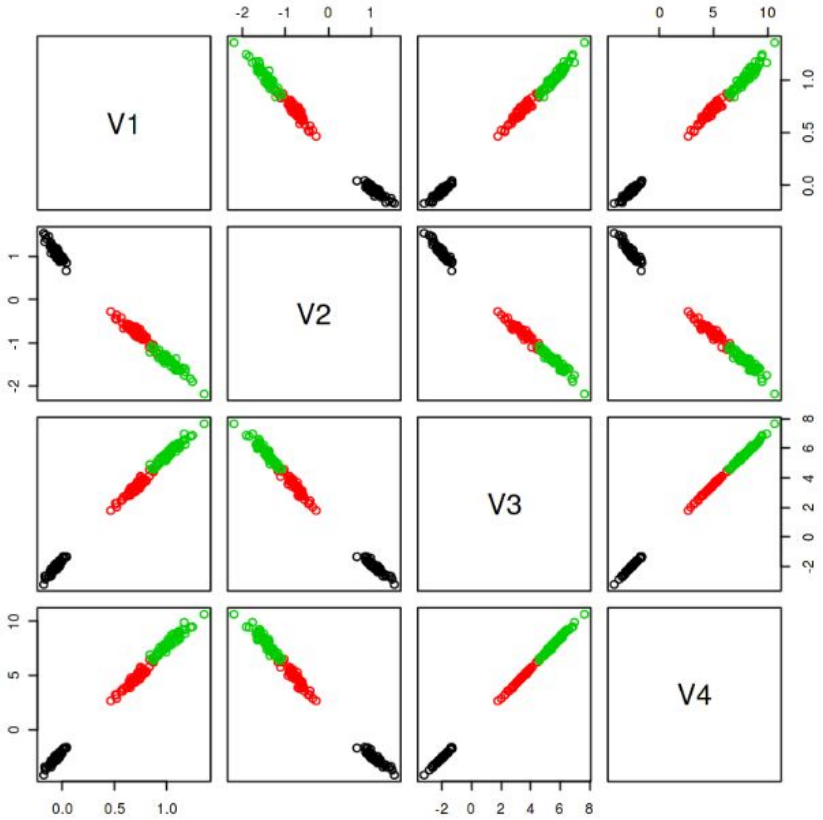
III. Neighbourhood Component Analysis results

1. Using Iris dataset

Neighbourhood component analysis was used for dimensionality reduction on iris dataset. The transformed data was then used for classification using a decision tree classifier.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1.0000	126.3000	-210.8346	-227.3737
-136.4336	-455.0854	545.8065	485.3440
655.1249	636.2119	-1792.2449	-1562.6671
854.3046	832.4459	-2259.9550	-2610.2199

The above table shows the transformed data after applying NCA.



The above plot shows a relation between V1, V2, V3, V4 which are the components obtained on applying NCA.

The confusion matrix is shown below-

Species	Setosa	Versicolor	Virginica
Setosa	20	0	0
Versicolor	0	19	1
Virginica	0	0	20

Accuracy obtained on classification using the decision tree is 98.33%..

2. Using Digits Dataset

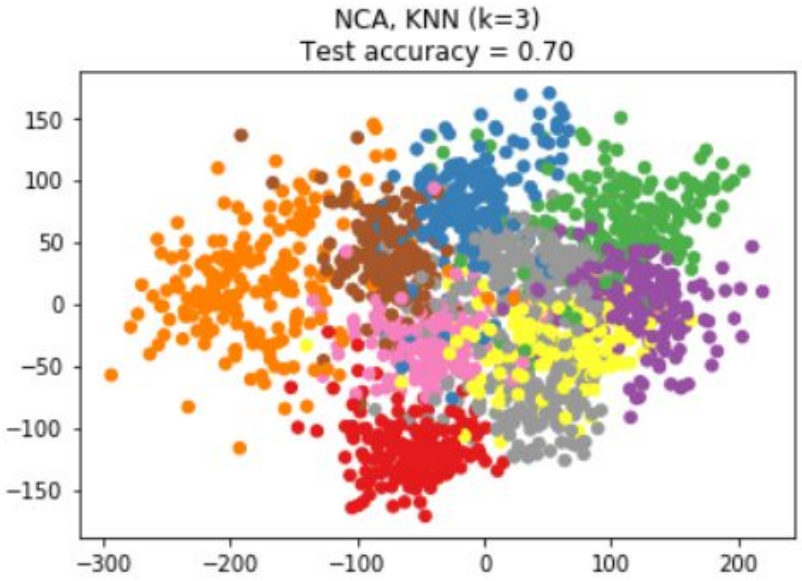
Neighbourhood component analysis tries to find a feature space such that a stochastic nearest neighbour algorithm will give the best accuracy.

We have used the Digits dataset to compare the classification accuracy on applying different dimensionality reduction techniques- NCA, LDA and PCA. We use K-Nearest Neighbours and decision trees for classification. The results obtained are described below.

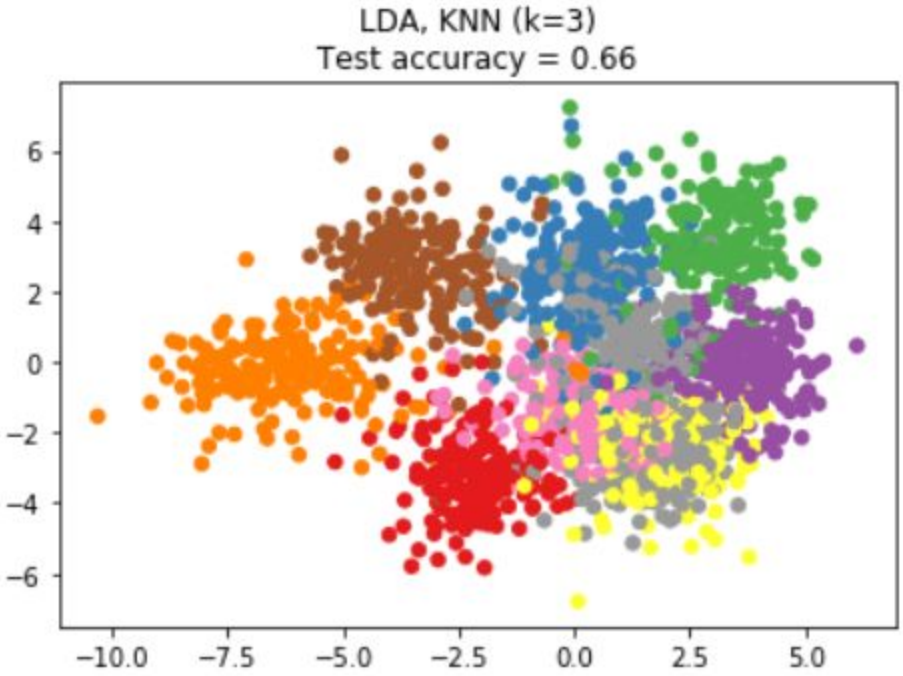
a. Using K-Nearest Neighbours (k=3)

For evaluation, the 3-nearest neighbor classification accuracy was computed on the 2-dimensional projected points found by each method.

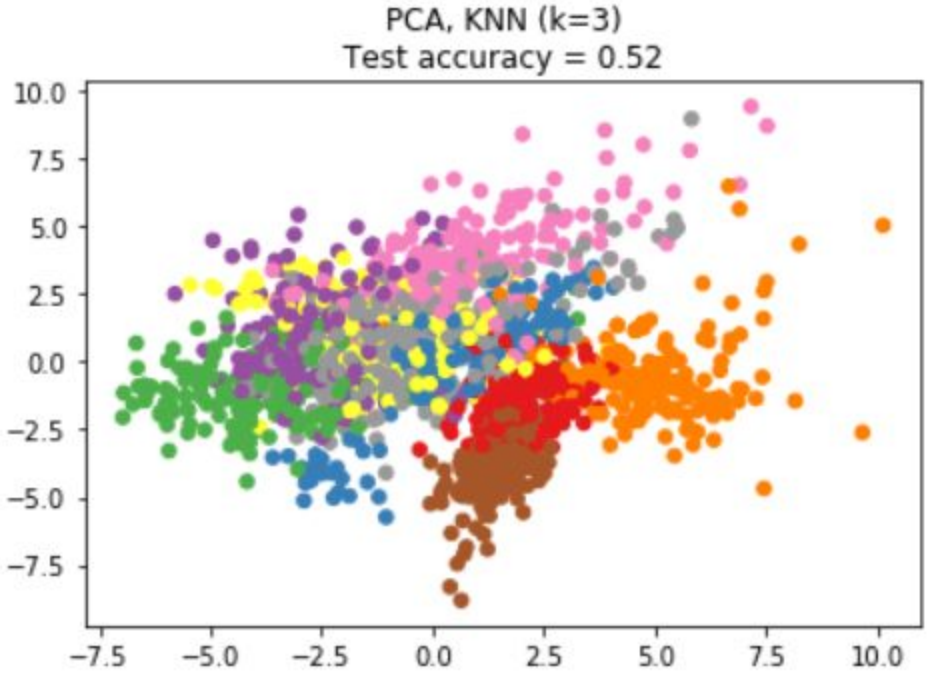
Accuracy of classification obtained after using NCA for dimensionality reduction to two components was 70%.



LDA was used to reduce the dimensionality of data and classification accuracy obtained was 66%.



We also performed PCA to reduce each image to a two-dimensional data point and classified using 3- nearest neighbors classifier.
Accuracy of classification obtained after PCA was 52%.



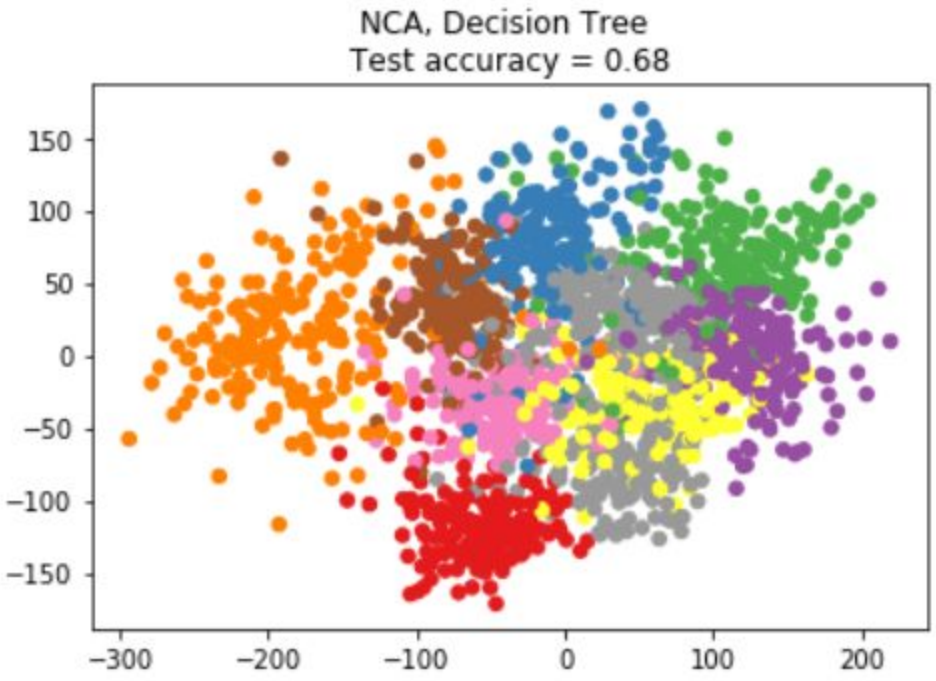
The result obtained is summarized in the following table-

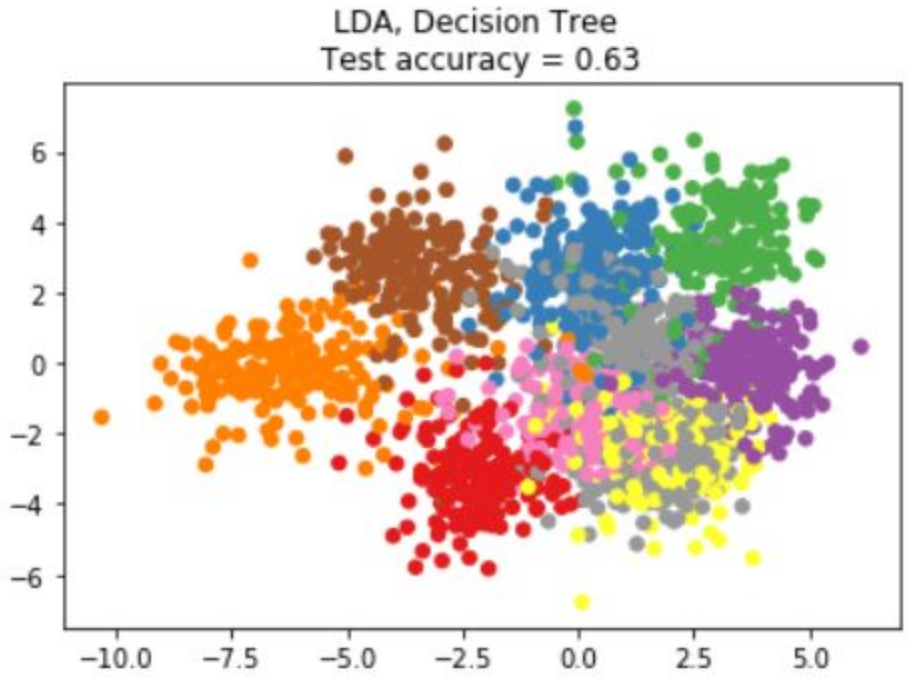
Dimensionality Reduction technique	**Accuracy of classification (using K-Nearest Neighbours, K=3)**
Neighbourhood Component Analysis	70%
Linear Discriminant Analysis	66%
Principal Component Analysis	52%

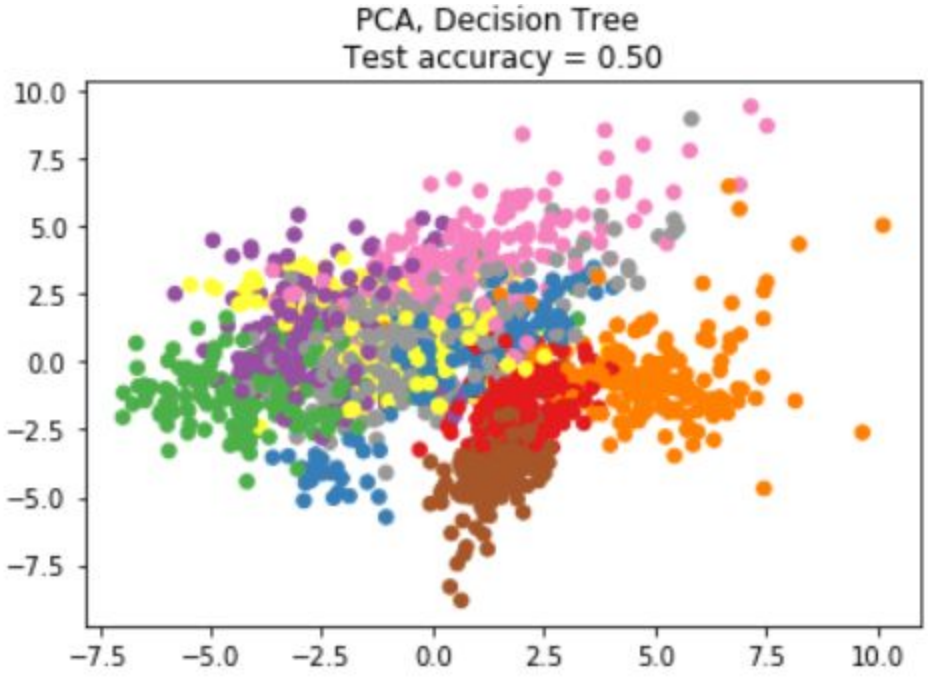
From Fig--(NCA), we observe that Neighbourhood Component Analysis is able to enforce clustering of data despite high dimensionality reduction from 64 dimensions to 2 dimensions.

b. Using decision tree classifier

We also used decision tree classifier to compare the classification accuracy for the dimensionally reduced data (64 dimensions to 2 dimensions) which was reduced by NCA, LDA and PCA.




LDA, Decision Tree
Test accuracy = 0.63
6
4
2
0
−2
−4
−6
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0


PCA, Decision Tree
Test accuracy = 0.50
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

The result obtained is summarized in the following table-

Dimensionality Reduction technique	**Accuracy of classification (using decision tree)**
Neighbourhood Component Analysis	68%
Linear Discriminant Analysis	63%
Principal Component Analysis	50%

From the above results, we can conclude that both the classifiers namely- K-Nearest Neighbour and Neighbourhood Component Analysis give the best classification accuracy when NCA is used for dimensionality reduction (in comparison with PCA and LDA). In general, we can also conclude that NCA and LDA which are supervised dimensionality reduction techniques, perform better in comparison to PCA which is unsupervised dimensionality reduction technique.

IV. Fisher's Linear Discriminant Results

1. Iris dataset

Since linear discriminant analysis can be affected by the scale in which feature variables are measured, we have normalized the continuous valued attributes before the analysis. We have divided the dataset into 75% training and 25% test data.

The LDA algorithm works by finding the directions that maximize the separation between classes. These directions are used to predict the class of data points. These directions, called linear discriminants. These are linear combinations of predictor variables. LDA makes the assumption that predictors are normally distributed and that the different classes have class-specific means.
The linear discriminant analysis is computed using the function lda() [MASS package].
LDA finds group means and computes the probability of each data point, belonging to the different groups. The lda() outputs presented below have the following information:

1) Group means: the mean of each variable in each group.
2) Coefficients of linear discriminants: the linear combination of predictor variables that are used in forming the LDA decision rule.

LD1 = 0.59*Sepal.Length + 0.77*Sepal.Width - 3.60*Petal.Length - 2.38*Petal.Width.
LD2 = -0.11*Sepal.Length -0.94*Sepal.Width +1.57*Petal.Length - 1.97*Petal.Width.

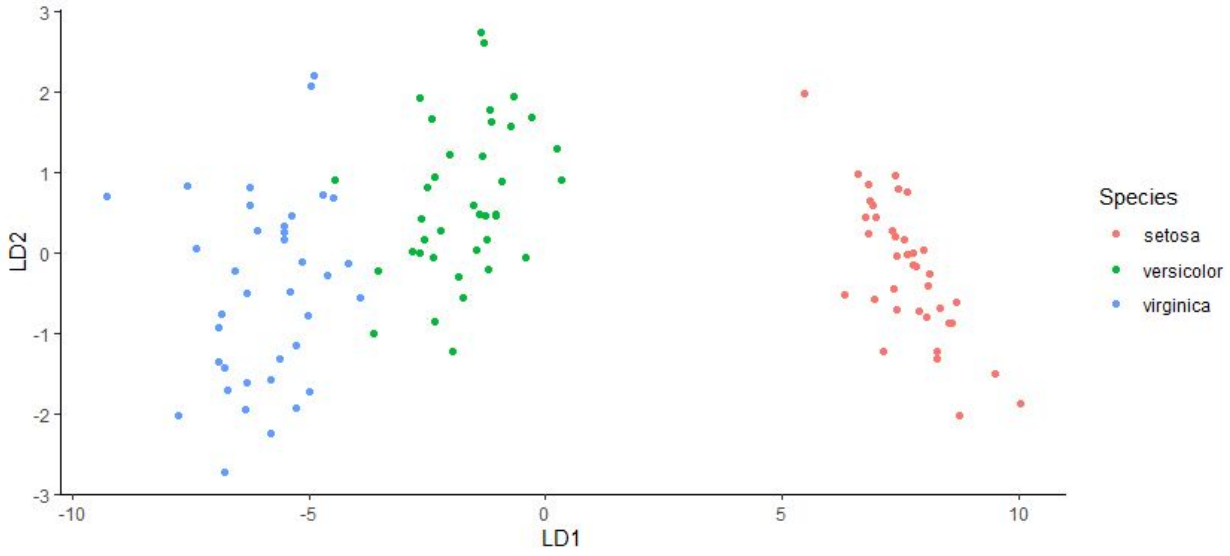
Coefficients of linear discriminants:
Group means:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
setosa	-0.99175999	0.7991714	-1.286096	-1.2491173
versicolor	0.08044049	-0.6215777	0.253091	0.1629779
virginica	0.91131950	-0.1775936	1.033005	1.0861394

Proportion of trace:

LD1	LD2
0.9931	0.0069

	LD1	LD2
Sepal.Length	0.5987383	-0.1142605
Sepal.Width	0.7684811	-0.9433342
Petal.Length	-3.6027055	1.5753489
Petal.Width	-2.3894022	-1.9732916



Model accuracy: 97.22222%

2. Glass Dataset

Prior probabilities of groups:

Type	Prior Probabilities
1	0.31443299
2	0.36597938
3	0.08247423
5	0.06185567
6	0.04123711
7	0.13402062

Group means:

Type	RI	Na	Mg	Al	Si
1	1.518536	13.21197	3.5275410	1.172131	72.66049
2	1.518645	13.10141	3.0259155	1.398310	72.59451
3	1.517941	13.44375	3.5562500	1.180000	72.42125
5	1.518712	12.97750	0.6958333	2.073333	72.27667
6	1.517216	14.75375	1.1675000	1.388750	73.26250
7	1.517143	14.40769	0.4750000	2.118846	73.00923

Type	K	Ca	Ba	Fe
1	0.4688525	8.779508	0.01311475	0.06016393
2	0.5174648	9.079014	0.04802817	0.08042254
3	0.3968750	8.770000	0.00937500	0.06062500
5	1.5441667	9.999167	0.20333333	0.06583333
6	0.0000000	9.305000	0.00000000	0.00000000
7	0.3065385	8.599615	1.01192308	0.01192308

Coefficients of linear discriminants:

	LD1	LD2	LD3	LD4	LD5
RI	-295.7855133	-65.46010525	374.5170163	114.9762459	-818.19365912
Na	-2.1358519	-3.44085954	1.0981478	6.9644377	1.56126475
Mg	-0.3509903	-3.39407666	2.2365515	6.8945777	1.92371559
Al	-2.9825400	-2.29919097	2.9898513	6.2410201	0.04862168
Si	-2.0651193	-3.53452002	2.4511278	7.4003016	0.10045521
K	-1.2524141	-2.27662756	2.0548943	8.1421800	1.85924189
Ca	-0.6952102	-2.75649056	1.3550874	6.8849544	2.82824061
Ba	-1.7724714	-3.91008445	3.2269437	6.5855868	3.63847804
Fe	0.2041211	-0.02964934	0.8497919	0.4281793	-0.94193927

Proportion of trace:

LD1	LD2	LD3	LD4	LD5
0.8184	0.1107	0.0451	0.0146	0.0112

Model accuracy: 0.75

3. Wine Dataset

Prior probabilities of groups:

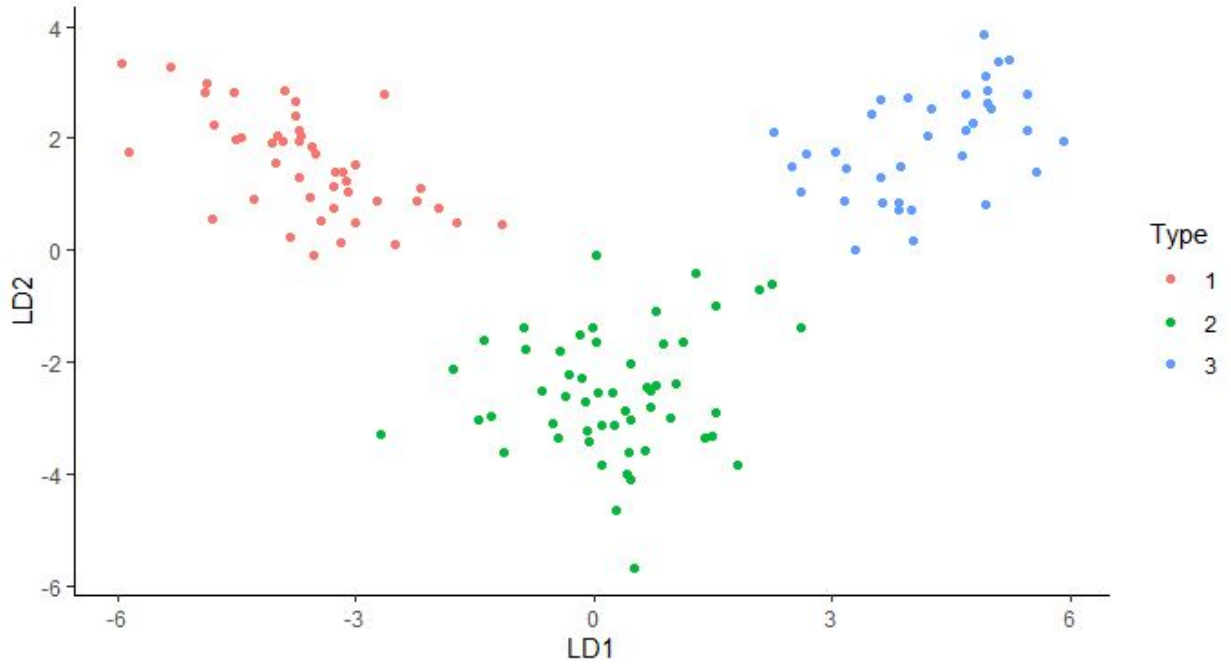
1	2	3
0.3333333	0.4000000	0.2666667

Group means:

Type	Alcohol	Malic	Ash	Alcalinity	Magnesium	Phenols
1	0.9202176	-0.2959190	0.3585114	-0.7802048	0.51903096	0.83126458
2	-0.8693210	-0.3771352	-0.4602177	0.2835218	-0.46665974	-0.03855374
3	0.1537095	0.9356016	0.2421873	0.5499733	0.05120091	-0.98125012

	Flavanoids	Nonflavanoids	Proanthocyanins	Color	Hue
1	0.91505434	-0.57274859	0.54755620	0.1867629	0.4784353
2	0.07210845	0.06812564	0.05542949	-0.8166154	0.4034576
3	-1.25198060	0.61374728	-0.76758949	0.9914695	-1.2032306

	Dilution	Proline
1	0.7500966	1.1872083
2	0.2519856	-0.7278734
3	-1.3155991	-0.3922003



Coefficients of linear discriminants:

	LD1	LD2
Alcohol	-0.40072815	0.62872362
Malic	0.23106291	0.47174832
Ash	-0.19526176	0.73587719
Alcalinity	0.62987389	-0.55625503
Magnesium	-0.20127030	-0.05033348
Phenols	0.46679617	0.08766805
Flavanoids	-1.50974545	-0.76566880
Nonflavanoids	-0.19352362	-0.22713659

Proanthocyanins	-0.07835612	-0.15117426
Color	0.91453044	0.72168213
Hue	-0.04242134	-0.23425433
Dilution	-0.76347653	0.01733591
Proline	-1.00127094	0.81991961

Proportion of trace:
LD1 LD2
0.6748 0.3252

Model accuracy: 0.9767442

Conclusion

In Section I of the Results section, PCA was performed on wine dataset and classification results were shown using decision tree classifier and logistic regression. From the results obtained, we could conclude that dimensionality reduction not only helped in better computation but also gave better evaluation results. In Section II, SVD was performed on iris dataset and decision tree classifier was used for evaluation. In Section III, NCA was performed on iris dataset and digits dataset. We compared the classification accuracy after dimensionality reduction using PCA, LDA and NCA on digits dataset. Based on the evaluation results on using K-Nearest Neighbours (k=3) and decision tree classifier, we concluded that NCA and LDA performed better than PCA. This is because NCA and LDA are supervised dimensionality reduction techniques unlike PCA which is unsupervised. We thus infer that supervised dimensionality reduction techniques perform better than unsupervised dimensionality reduction techniques. In Section IV, we performed Fisher's Linear Discriminant analysis on three datasets namely- Iris dataset, wine dataset and glass dataset.

References

1. https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/princomp
2. http://people.bu.edu/bkulis/pubs/ftml_metric_learning.pdf
3. https://link.springer.com/article/10.1007/s10618-019-00616-4
4. https://www.cs.toronto.edu/~hinton/absps/nca.pdf
5. https://www.ics.uci.edu/~welling/teaching/273ASpring09/Fisher-LDA.pdf
6.https://www.sciencedirect.com/topics/medicine-and-dentistry/principal-component-analysis
7. https://arxiv.org/abs/1812.05944
8. https://ieeexplore.ieee.org/document/7952041
9. https://www.sciencedirect.com/topics/computer-science/decision-trees
10. https://www.rdocumentation.org/packages/rpart/versions/4.1-15/topics/rpart
11. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4916348/