University of Cape Town

Department of Statistical Sciences

STA5077Z Unsupervised Learning

Assignment 2

EVRCHA001 - Charl Everts

20 November 2020

Department of Statistical Sciences

Plagiarism Declaration Form

A copy of this form completed and signed, to be attached to all coursework submissions to the Statistical Sciences Department. Submissions without this form will not be marked.

COURSE CODE: STA5077Z

COURSE NAME: Unsupervised Learning

STUDENT NAME: Charl Everts

STUDENT NUM: EVRCHA001

Plagiarism Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used a generally accepted citation and referencing style. Each contribution to, and quotation in, this tutorial/report/project from the work(s) of other people has been attributed and has been cited and referenced.
3. This tutorial/report/project is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.
5. I acknowledge that copying someone else's assignment or essay, or part of it, is wrong, and declare that this is my own work.

Note that agreement to this statement does not exonerate you from the University's plagiarism rules (http://www.uct.ac.za/uct/policies/plagiarism_students.pdf).

Signature:  Date: 20 November 2020

Table of Contents

List of Figures

List of Tables

Abbreviations/Symbols

FNA	Fine Needle Aspirate
ISOmap	Independent System Operator Map
KNN	K-Nearest Neighbour
MDS	Multi-Dimensional Scaling
MNIST	Modified National Institute of Standards and Technology
PCA	Principal Component Analysis
WDBC	Wisconsin Diagnostic Breast Cancer

Abstract

This report evaluates the improvement in model accuracy due to different dimensionality reduction techniques. A KNN (K-nearest neighbour) model is used to predict and classify the response variables in the MNIST and WDBC datasets: the number 5 or 8, and benign or malignant respectively. The first KNN model is trained on both full datasets to establish a benchmark accuracy. After dimensionality reduction is conducted, the algorithm is run again and the updated accuracy is compared to the benchmark value. The techniques evaluated are Principal Component Analysis (PCA), Metric Multi-Dimensional Scaling (MDS) and the Independent System Operator Map (ISOmap).

The explicit goals of the report are as follows:

i. Pre-process the data and use the KNN algorithm to classify handwritten digits and breast cancer diagnosis to determine a benchmark accuracy.

ii. Perform dimensionality reduction on the data using PCA, MDS and ISOmap techniques.

iii. Recommend a dimensionality reduction technique to use for each dataset based on the accuracy of the new KNN models built with reduced datasets.

iv. Produce a summary of the analysis and conclusions.

Chapter 1

Data Pre-Processing

1.1 The Dataset

The MNIST database (Modified National Institute of Standards and Technology) is an extensive database consisting of handwritten digits commonly used for training image processing systems and used in machine learning models. The digits are written by high school students and employees of the United States Census Bureau. The full database contains 60,000 training images and 10,000 testing images. The dataset used in this report contains only 10,000 images, split exactly in half between the digits 5 and 8 (5,000 each). Every row consists of a greyscale image of digit 5 or 8 represented by a one-dimension vector of size 785 columns. The first number of each row is the label of the image. The following 784 numbers are the pixels of the 28 x 28 image whose values range from 0 to 255.

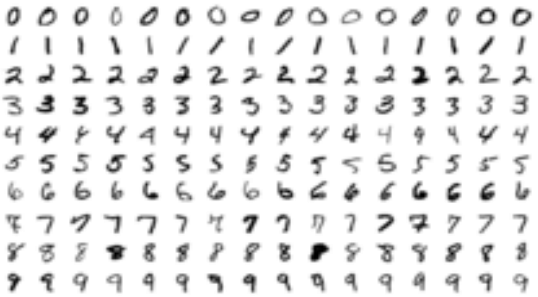

Figure 1.1: *Sample of handwritten digits from the full dataset.*

The Wisconsin Diagnostic Breast Cancer (WDBC) dataset consists of 569 data points classified as either malignant or benign. Data was computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. Each instance contains 30 features describing different characteristics of the cell nuclei present in the image.

1.2 Data Quality Assessment

Both datasets are well constructed with no missing values. The only column removed was the `id` column from the WDBC dataset as its only purpose was to serve as an index for the row entry. Since the WDBC dataset contained 30 distinct numeric features, all columns required normalising before implementation in the KNN algorithm. In MNIST however, normalising was not required since all the features were integers relating to the same grey-scale variable.

Figure 1.2 plots the standard deviation of features for the MNIST dataset. More than 200 features had zero variances and required removal before building the KNN model. These features corresponded to the columns which had repeated zeros since they were located on the perimeter of the image and hence never had a grey-scale value. This issue was not present in the WDBC dataset.

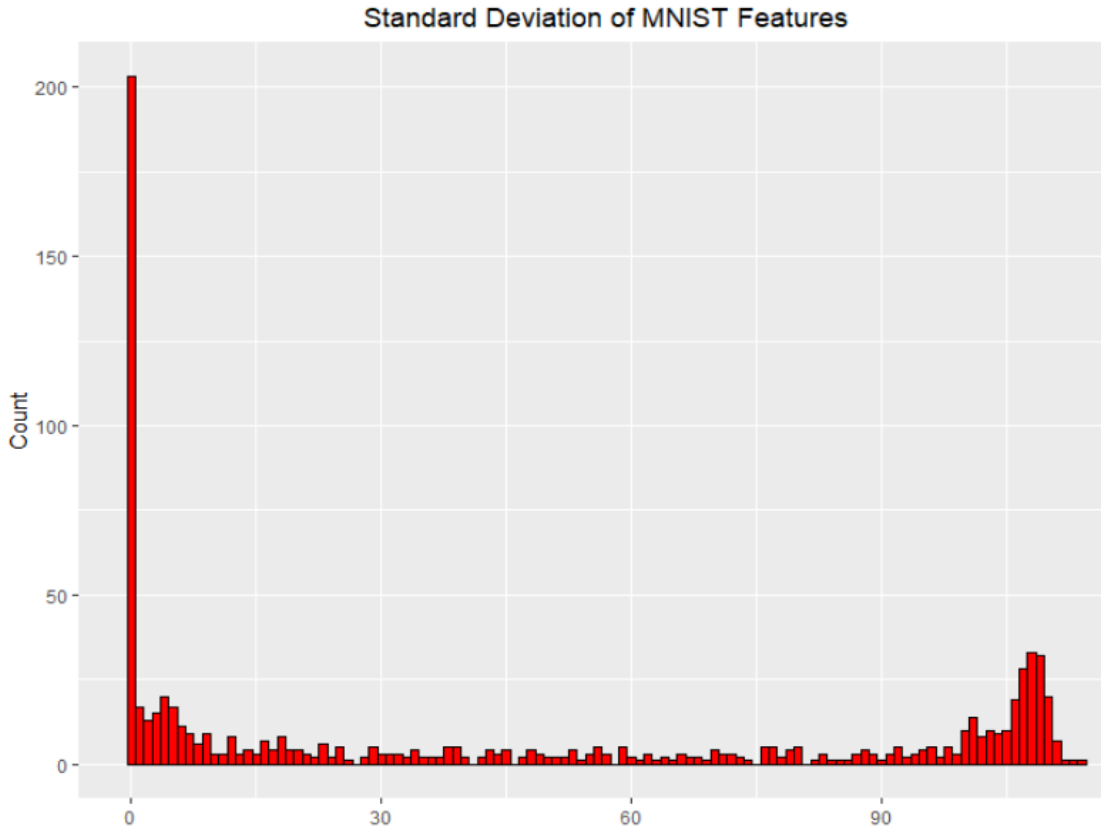

Figure 1.2: *Distribution of standard deviation values across MNIST dataset.*

Chapter 2

Methodology

2.1 KNN

K-nearest neighbour (KNN) is a primitive algorithm in which each observation is predicted based on how similar it is to other observations. It is a memory-based algorithm and cannot be summarized by a closed-form model. This means the training samples are required at run-time and predictions are made directly from the sample relationships. KNN models are therefore said to be *lazy learners* can be computationally inefficient. Euclidean distance is the most common measure of similarity and measures the straight-line distance between two samples. The KNN algorithm's process can be summarised as follows:

1. Specify the train and test data.

2. Select a value of "k" neighbours, i.e. the nearest data points.

3. For each point in the test data:
   - 3.1 Calculate the distance between test data and each row of training data using Euclidean distance.
   - 3.2 Add the distance and the index of the entry to an ordered collection.
   - 3.3 Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances.
   - 3.4 Pick the first k entries from the sorted collection.
   - 3.5 For this classification exercise, return the mode of the k labels.

Cons of KNN:

- It is computationally expensive as it requires storing all the training data (lazy learner).
- High memory storage required compared to other supervised learning algorithms.
- Prediction slows down greatly as data volume increases.
- Faster alternatives exist.
- Highly sensitive to the scale of data and irrelevant features.

Pros of KNN:

- Simple algorithm to understand and interpret.
- Useful for nonlinear data since no assumption is made about data types.
- Applicable to classification and regression problems.
- Has relatively high accuracy.
- Algorithm is unbiased easy to implement.
- KNN may perform well on big datasets provided you have sufficient computing resources.

As mentioned previously, the features with zero variance were removed from the dataset before building the KNN model on the respective datasets. An 80/20 percent split was done to create the training/testing split for each model. The `trainControl` function was used to conduct 10-fold cross-validation. The number of "k" neighbours to use in each model was decided by Figures 2.1 & 2.2

2.1.1 KNN for the WDBC dataset

The KNN model used for the benchmark classification accuracy had the highest accuracy for k = 3 neighbours (Figure 2.1). The respective sensitivity, specificity and accuracy values are displayed in Table 2.1.

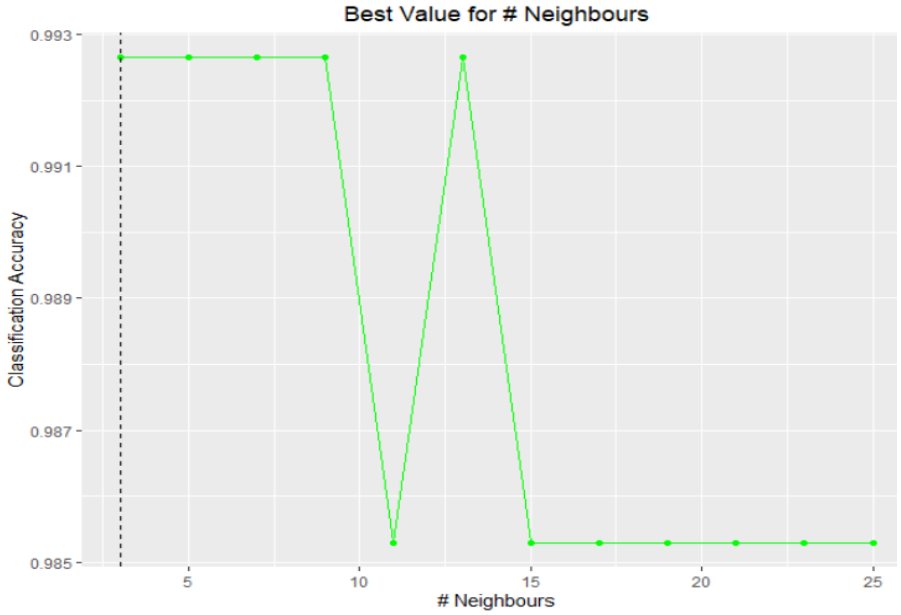

Figure 2.1: *Number of neighbours selected verses classification accuracy (WDBC).*

Classification Accuracy Benchmark for WDBC		
Sensitivity	Specificity	**Accuracy**
97.18	88.10	**92.64**

Table 2.1: *Classification accuracy for full WDBC dataset.*

2.1.2 KNN for the MNIST dataset

The KNN model used for the benchmark classification accuracy had the highest accuracy also for k = 3 neighbours (Figure 2.2). The respective sensitivity, specificity and accuracy values are displayed in Table 2.2.

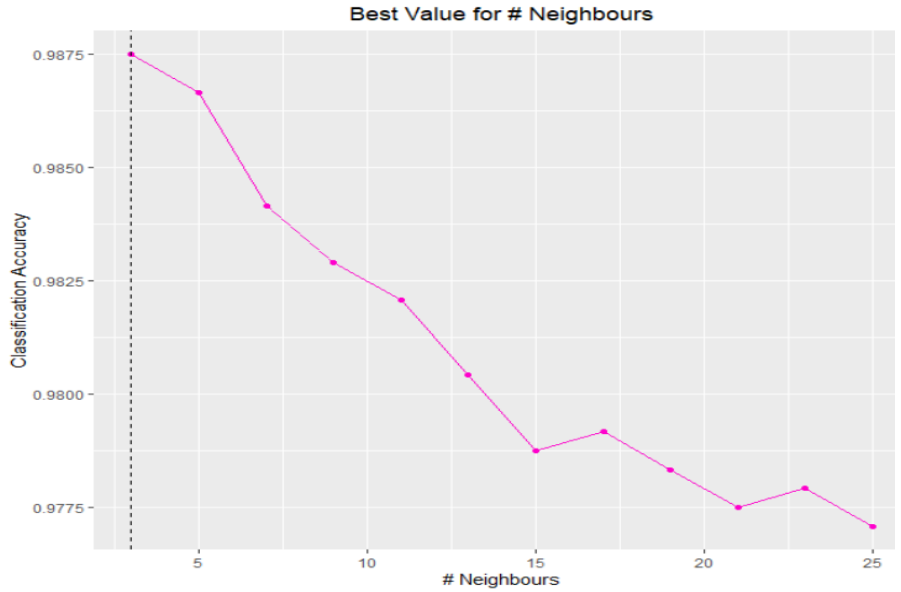

Figure 2.2: *Number of neighbours selected verses classification accuracy (MNIST).*

The standard KNN model using both full datasets had better results in predicting the outcome on the MNIST dataset. The MNIST KNN model surpassed all three WDBC sensitivity, specificity and accuracy levels.

Classification Accuracy Benchmark for MNIST		
Sensitivity	Specificity	**Accuracy**
99.80	97.70	**98.75**

Table 2.2: *Classification accuracy for full MNIST dataset.*

2.2 PCA

Principal Components Analysis (PCA) is an algorithm that transforms the variables in a dataset into a new set of features called principal components. It is a statistical procedure to describe a set of multivariate data of possibly correlated variables by relatively few numbers of linearly uncorrelated variables. The uncorrelated variables are a linear combination of the original variables and in the decreasing order of importance so that the first component explains most of the original variation in the data.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trade-off exists between accuracy and efficiency. In doing so, a large proportion of the dataset is effectively compressed into fewer features. This enables dimensionality reduction and ability to visualize the separation of clusters previously unknown due to high dimensionality.

Since the objective of PCA is to retain as few dimensions as possible with as much of the original information, it is very important to find out the optimized number of components to keep. The two methods used in this report are as follows:

1. Cumulative proportion of variance explained: generally, the number of components that explain 75-80% of the variance is chosen as the optimized number of components to be used.

2. Scree plot of eigenvalues: A graph of the eigenvalue and the number of components is made and a natural breakpoint is one where the slope of the graph is steep to the left of the breakpoint and not as steep to the right.

2.2.1 PCA for the WDBC dataset

Figure 2.3 is a scree plot depicting the number of theoretical principal components used against the associated change in variance. For eigenvalues<1 (variance axis), implies that the component explains less than a single explanatory variable. Figure 2.4 is a cumulative variance plot. Together these figures depict the first 6 components have an eigenvalue>1 and explain almost 90% of the variance. It is possible to effectively reduce dimensionality from 30 to 6 while only sacrificing 10% of variance. Figure 2.5 illustrates that the first 2 components explain more than 60% of the variance.

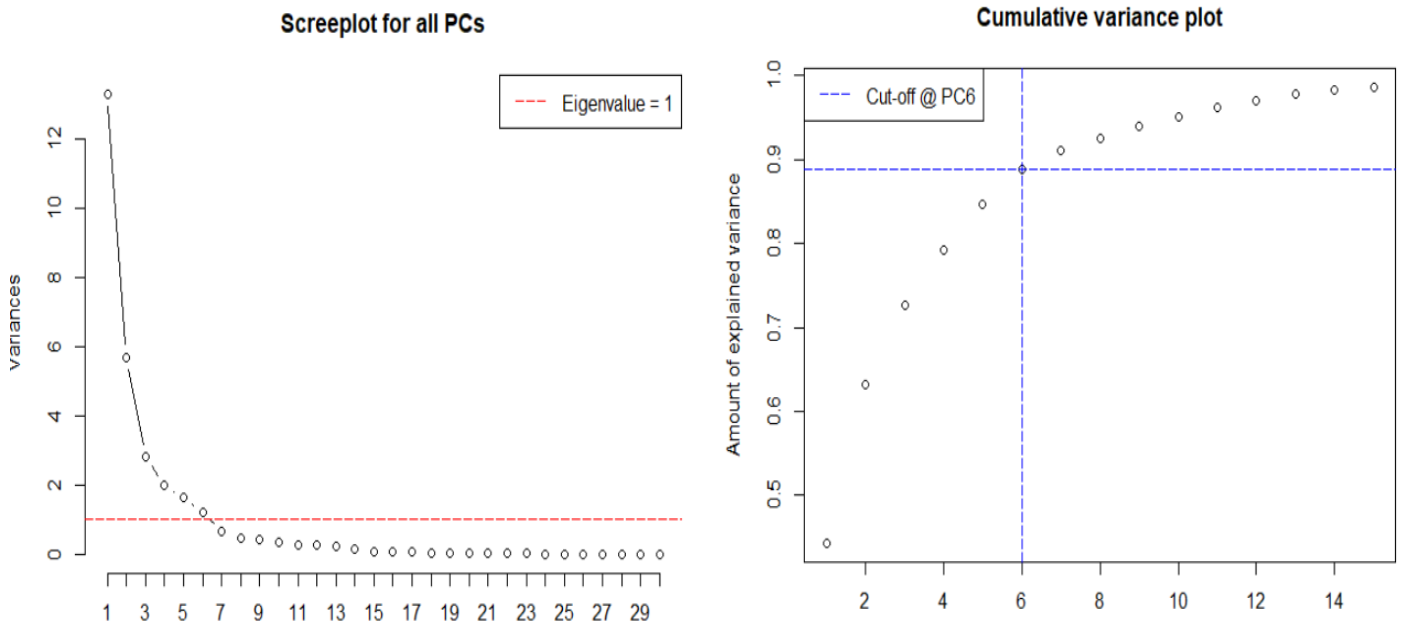

Figure 2.3 & 2.4: *Scree plot and Cumulative variance plot for all principal components.*

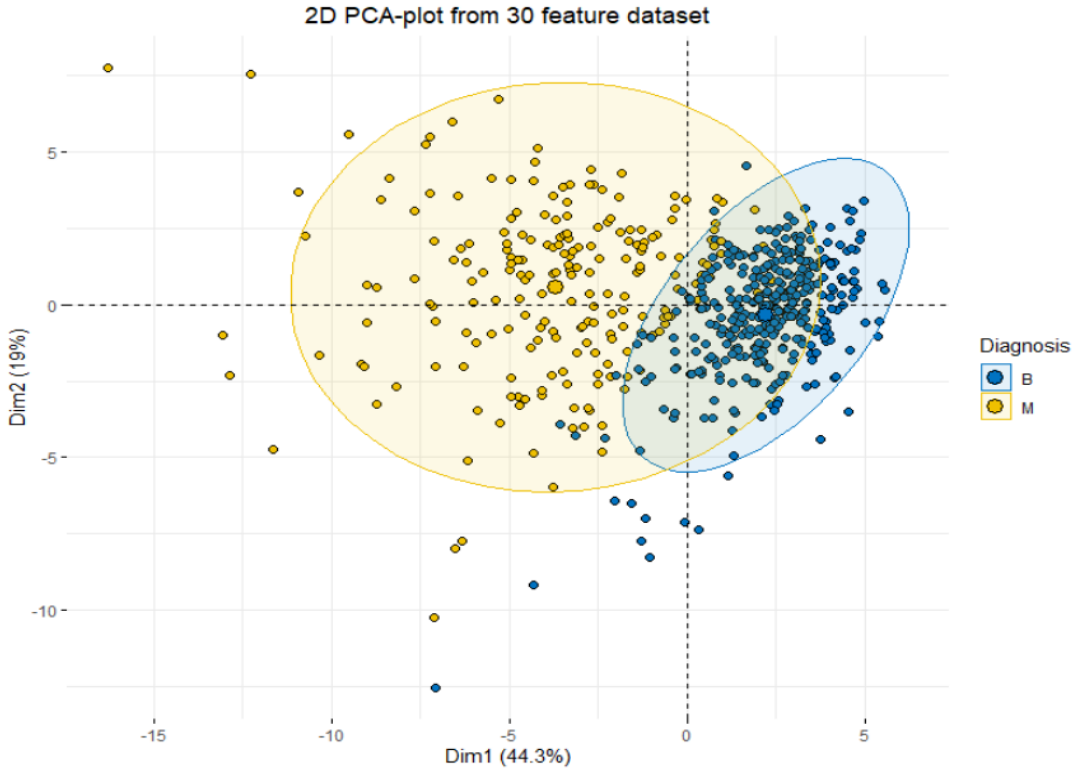

Figure 2.5: *2D PCA-plot from 30 feature WDBC dataset.*

Figure 2.6 visualizes the contribution of all 30 dataset features to the first 2 dimensions. Yellow has the least contribution while green contribute the most.

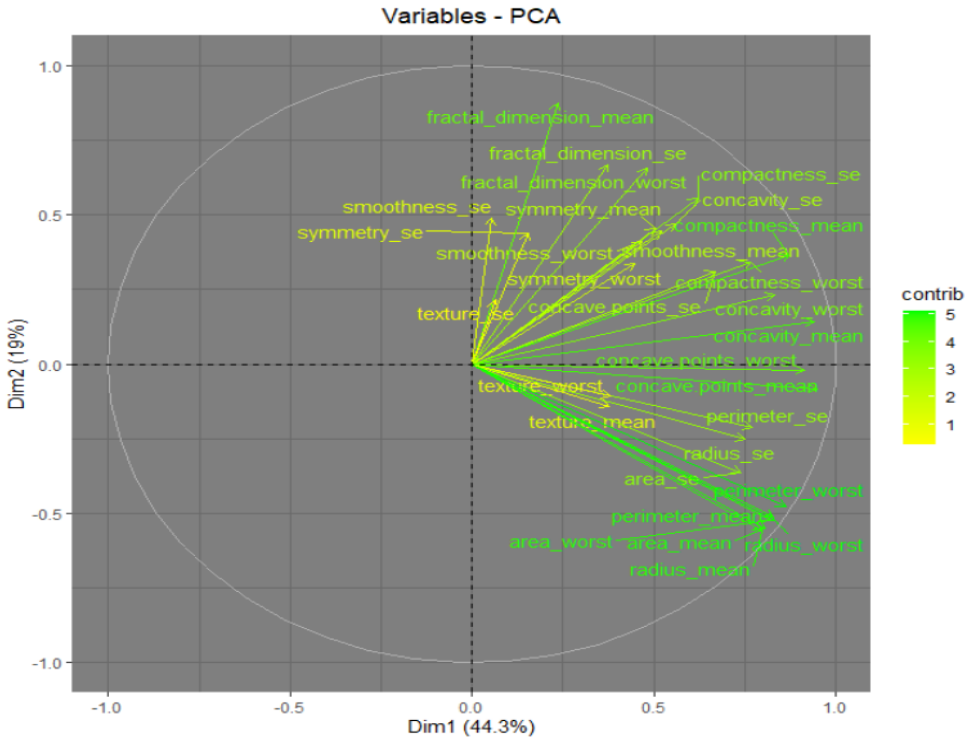

Figure 2.6: *The contribution of each feature to the top 2 dimensions.*

2.2.2 PCA for the MNIST dataset

The same process is repeated for the MNIST dataset, however, far more dimensions are required to explain an acceptable level of variance. In the WDBC PCA, 6 dimensions explained 90% of the variance, while in MNIST you need 42 dimensions to explain 80% of the variance. This is probably because MNIST has almost 30 times the number of features than WDBC.

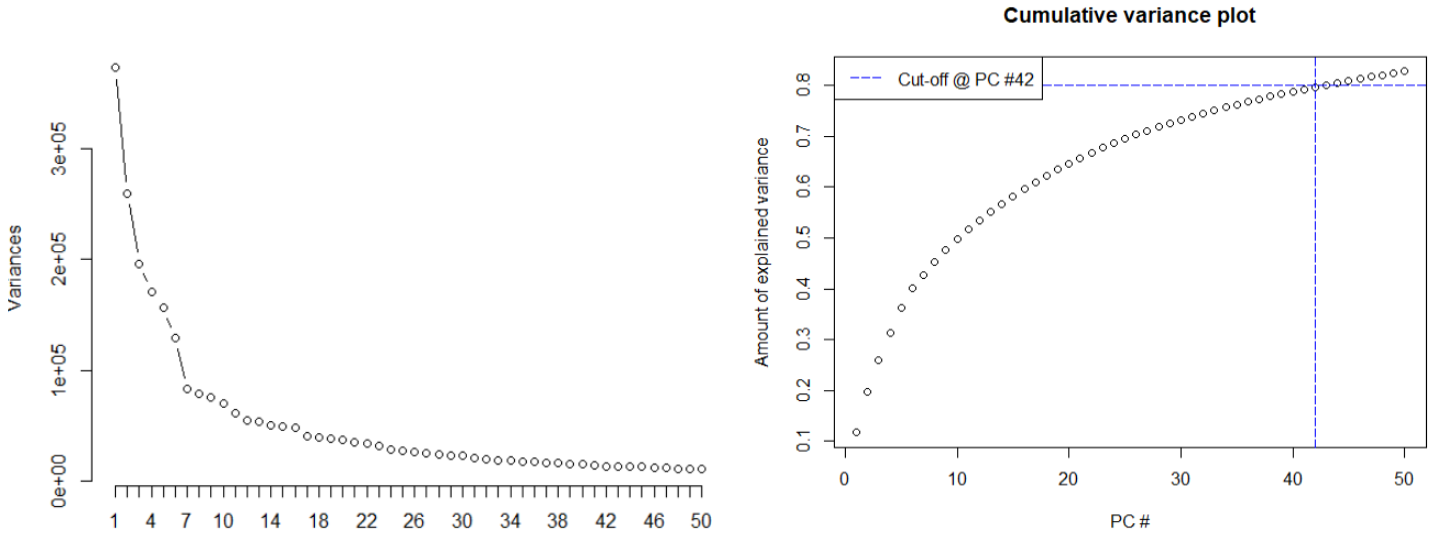

Figure 2.7 & 2.8: *Scree plot and Cumulative variance plot for all principal components.*

As evident in Figure 2.9, it is much harder to visualise the MNIST dataset due to the volume of observations. It is therefore infeasible to recreate an equivalent version of Figure 2.6. For the MNIST PCA, the first 2 dimensions explain less than 20% of the variance, an issue which may relate to the sheer size of the dataset.

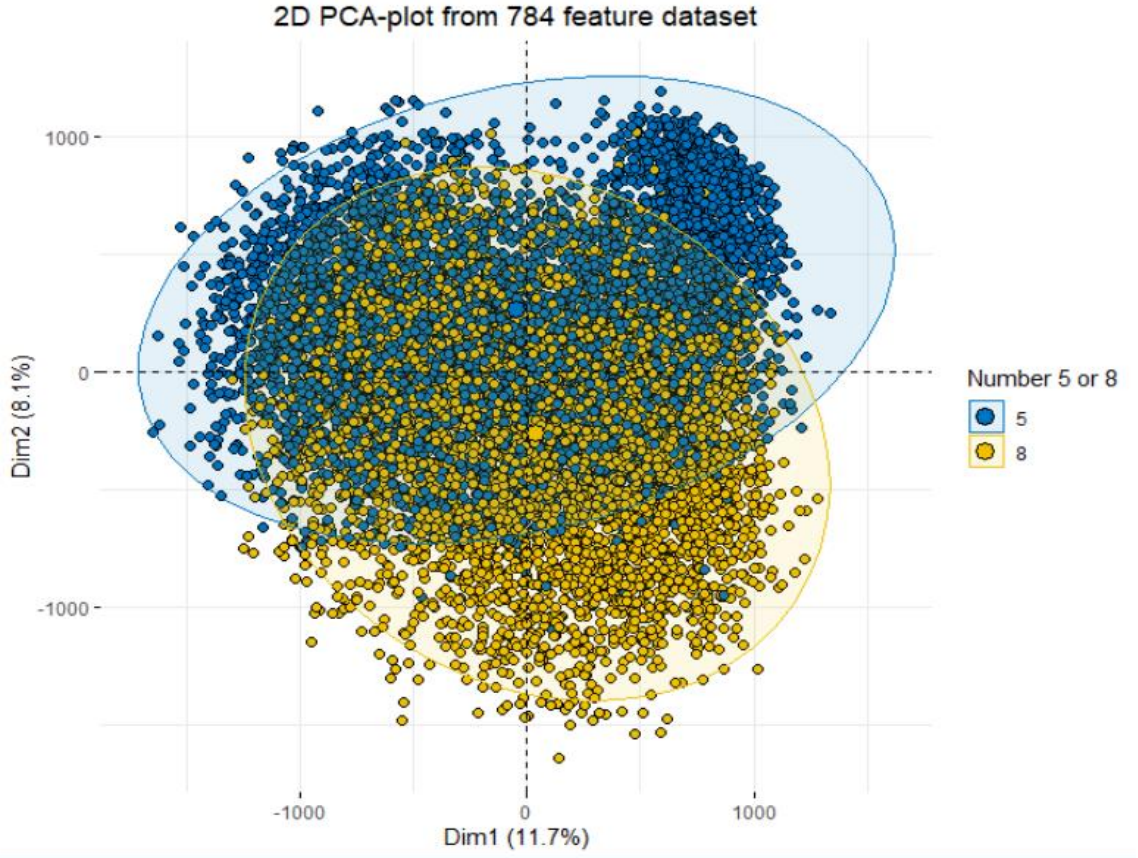

Figure 2.9: *2D PCA-plot from 784 feature WDBC dataset.*

2.3 Metric MDS

Multidimensional scaling (MDS) is a technique that creates a map displaying the relative positions of several objects, given only a table of the distances between them. The map may consist of multiple dimensions, or only a few. The table of distances is known as the distance matrix. It arises either directly from experiments or indirectly as a correlation matrix. The algorithm offers two methods: metric and non-metric MDS. The metric version attempts to reproduce the original distances, while non-metric assumes that only the ranks of the distances are known. Only the metric MDS algorithm is considered in this report.

2.3.1 Metric MDS for the WDBC dataset

To reduce the dimensionality of this dataset, a distance matrix is computed across all features before building the MDS model using the `cmdscale` function. Figure 2.10 plots the mapping of the MDS representation of the pairwise distance across the WDBC features.

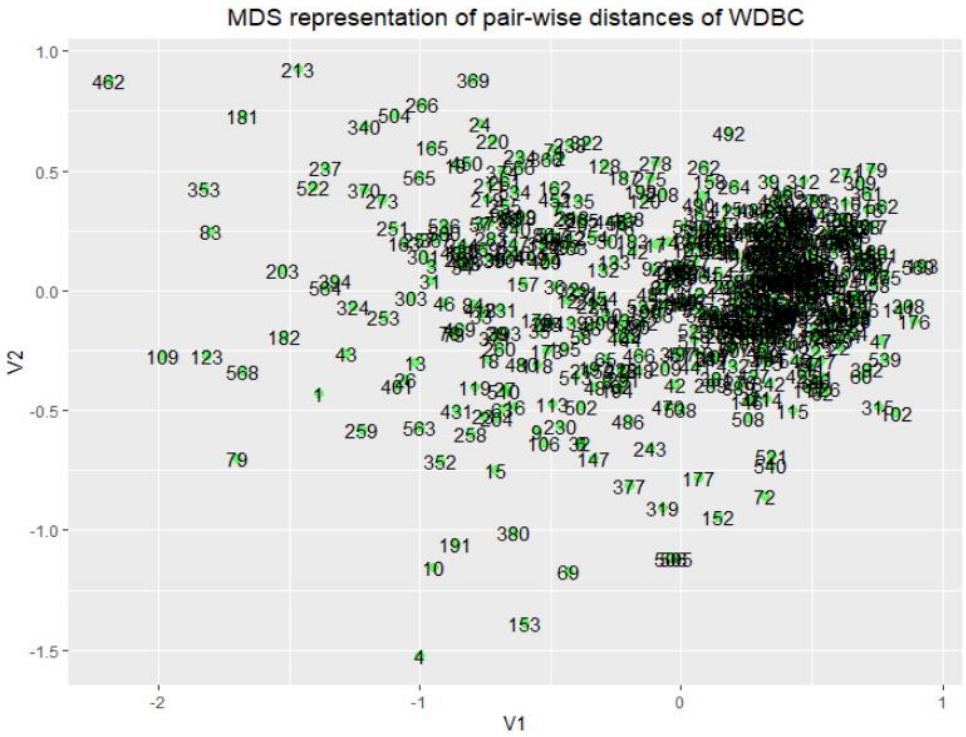

Figure 2.10: *Metric MDS representation of pairwise distance of WDBC dataset.*

2.3.1 Metric MDS for the WDBC dataset

The same process is repeated for the MNIST dataset, however, the visualisation of the MDS representation is rather pointless as it displays 20,000 points in one big cluster.

2.4 ISOmap

ISOmap stands for isometric mapping and is a nonlinear dimension reduction technique that preserves global properties of the dataset in question. Subsequently, the geodesic distances between all samples are captured best in the low dimensional embedding. The `Isomap` function in R version is based on the Matlab implementation by Tenenbaum and uses Floyd's Algorithm to compute the neighbourhood graph of shortest distances when calculating the geodesic distances. To estimate the intrinsic dimension of the data, the function plots the residuals between the high and low dimensional data for a given range of dimensions. It has been established that ISOmap does not perform well on large datasets.

2.4.1 ISOmap for the WDBC dataset

Figure 2.11 plots the decrease in residual variance, hence increase in accuracy of the ISOmap algorithm as the number of dimensions increase. The best accuracy is achieved the 30-feature dataset is reduced to 6 dimensions.

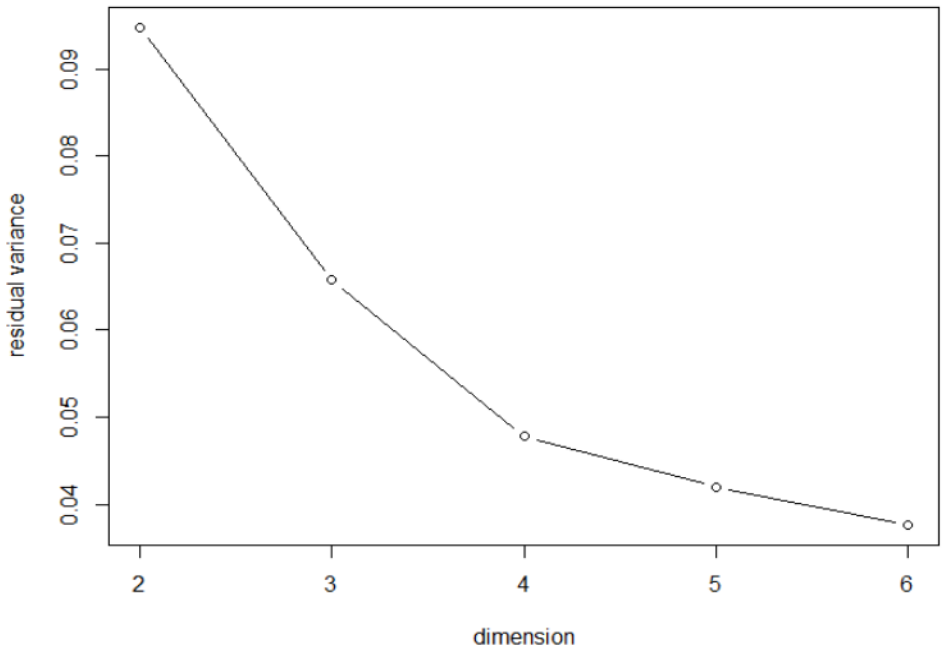

Figure 2.11: *Residual variance plotted against number of ISOmap dimensions used.*

2.4.2 ISOmap for the MNIST dataset

ISOmap failed to successfully reduce the MNIST dataset. As mentioned, the algorithm is notoriously inefficient when working with big datasets. The algorithm was run several times, and either crashed the program, yielded a run time error or was aborted after more than an hour of computation time. If we were to apply this method on the entire MNIST dataset containing 60,000 observations, on a standard machine it could result in more than 12 hours in run time. The algorithm was therefore deemed inappropriate to use on the MNIST dataset.

Chapter 3

3.1 Results

The results for both WDBC and MNIST datasets are given in Tables 3.1 & 3.2. The metric MDS and ISOmap algorithms shared the best performance on the WDBC dataset, yielding a classification accuracy of 94.69%. PCA comfortably outperformed metric MDS and ISOmap on the MNIST dataset, yielding a classification accuracy of 99.25%

KNN Classification Accuracy for Reduced WDBC			
Reduction Method	Sensitivity	Specificity	**Accuracy**
PCA	97.18	85.71	**92.92**
Metric MDS	98.59	88.10	**94.69**
ISOmap	98.59	88.10	**94.69**

Table 3.1: *KNN Classification accuracy for reduced WDBC datasets.*

KNN Classification Accuracy for Reduced MNIST			
Reduction Method	Sensitivity	Specificity	**Accuracy**
PCA	99.60	98.90	**99.25**
Metric MDS	96.90	97.00	**97.00**
ISOmap	**-**	-	**-**

Table 3.2: *KNN Classification accuracy for reduced MNIST datasets.*

3.2 Conclusion

Table 3.3 combines the benchmark and reduced dataset accuracies for each dataset and dimensionality reduction method applied. The poorest performing algorithm was ISOmap on MNIST. It failed to yield a result in more than an hour run time. Metric MDS did not improve the classification accuracy, however, PCA increased the accuracy by 0.50%. All three techniques used on the WDBC dataset improved the classification accuracy. The best method being Metric MDS and ISOmap which improved classification by more than 2%.

Dataset	**Reduction Method**	**Benchmark Accuracy**	**Reduced Accuracy**
WDBC	PCA	**92.64**	**92.92**
	Metric MDS		**94.69**
	ISOmap		**94.69**
MNIST	PCA	**98.75**	**99.25**
	Metric MDS		**97.00**
	ISOmap		**-**

Table 3.3: *Final summary respective benchmark and reduced dataset accuracies.*

The power of these dimensionality reduction techniques is evident when because not only was the dataset made more compact and manageable, the size reduction resulted in an increase in classification accuracy in 4 out of the 6 instances.

Appendix

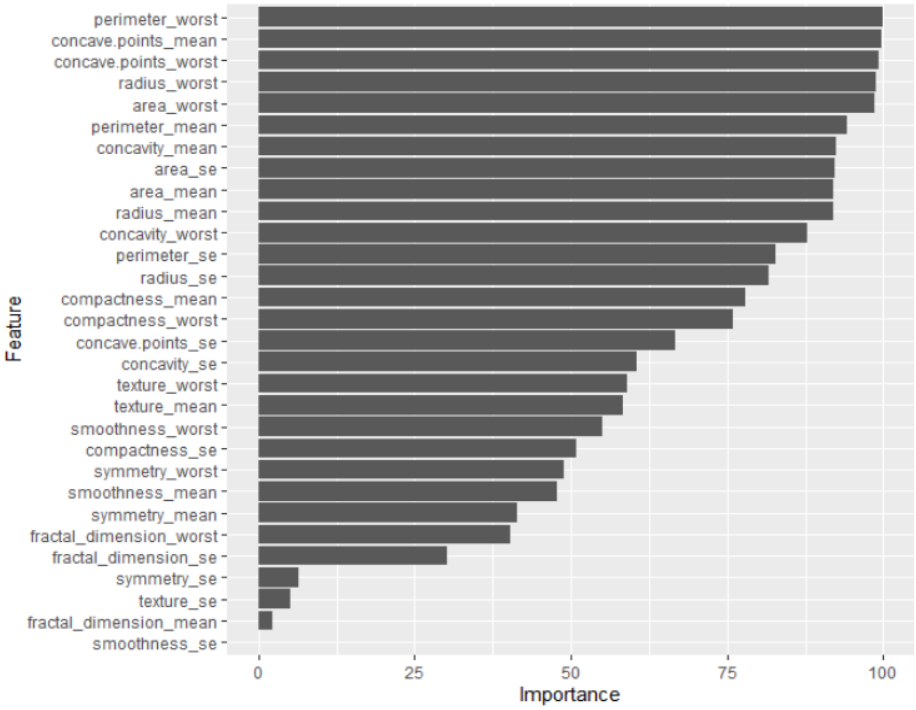

Figure A.1: *Variable importance of WDBC features for full dataset.*

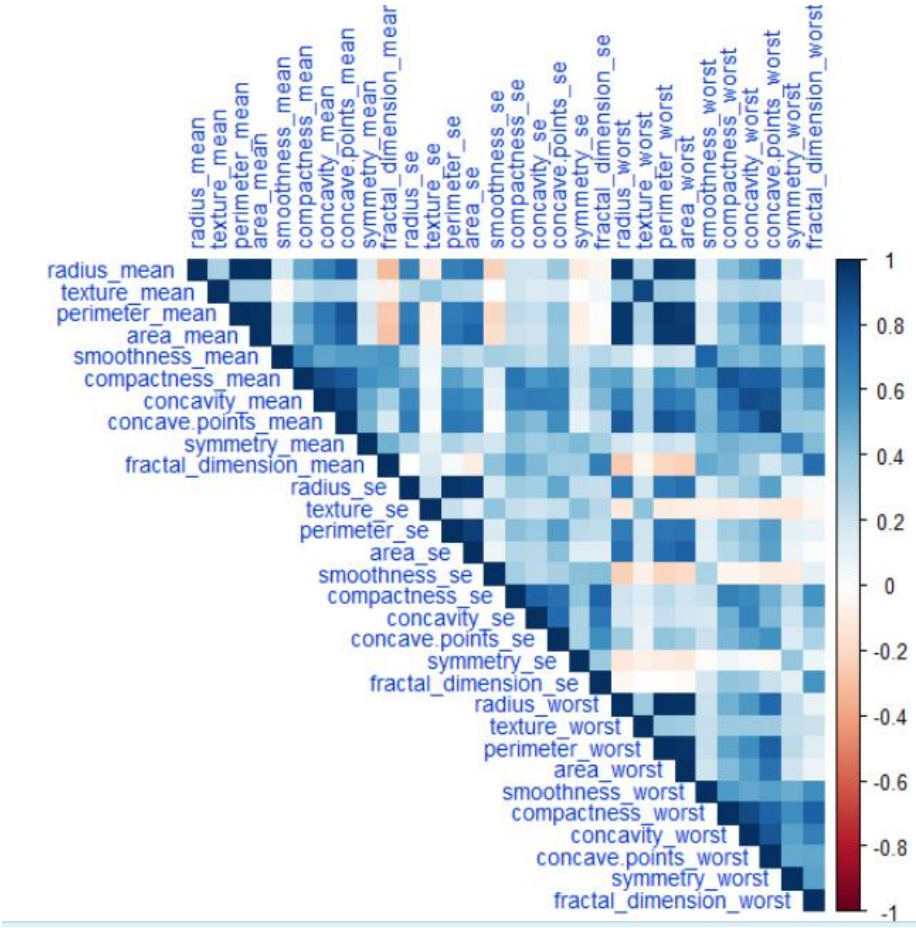

Figure A.2: *Correlation heatmap for WDBC dataset.*

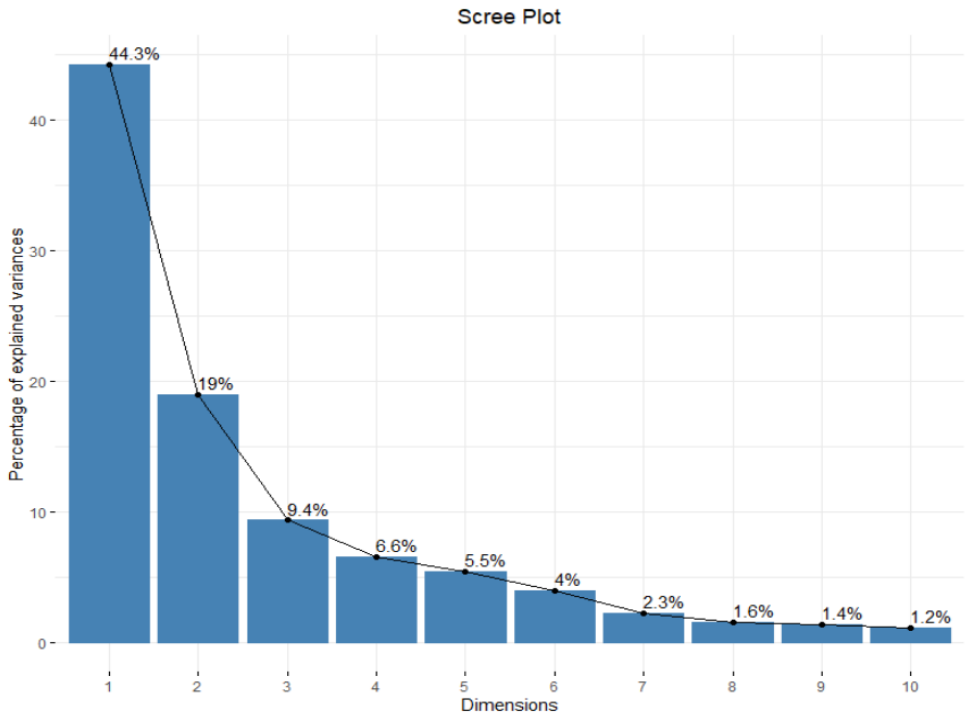

Figure A.3: Scree plot depicting the amount of variance explained by each additional dimension in WDBC PCA.

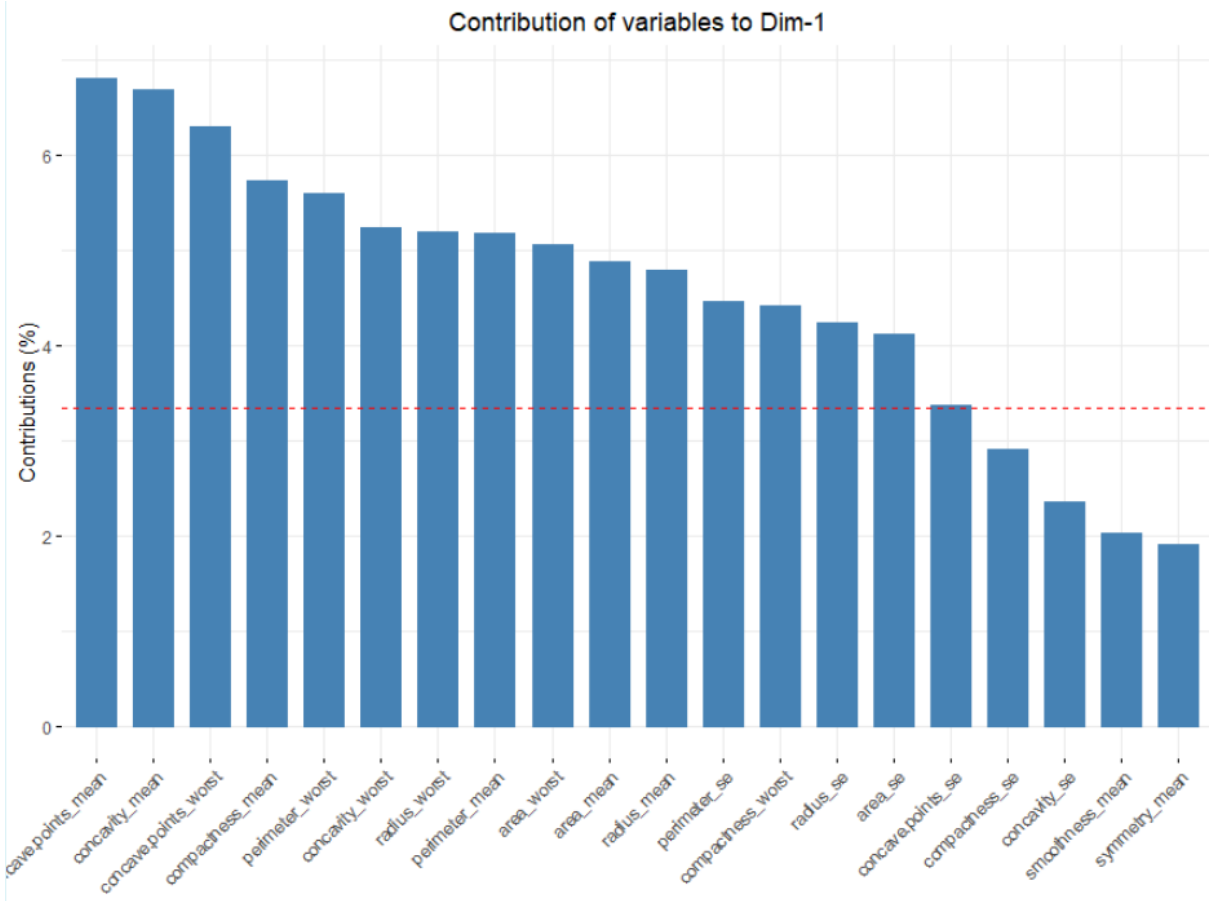

Figure A.4: *Contributions to dimension-1 by feature.*

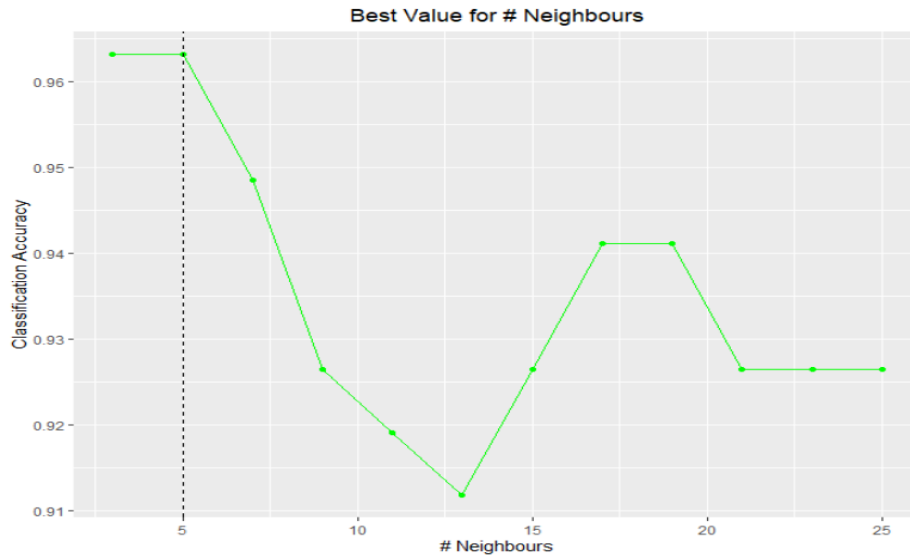

Figure A.5: *Nearest neighbour plot for reduced WDBC by PCA.*

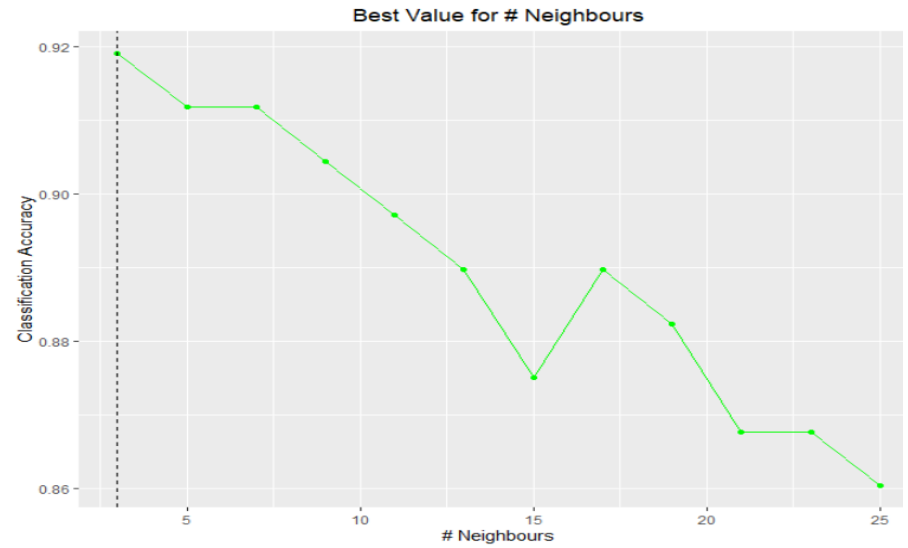

Figure A.6: *Nearest neighbour plot for reduced WDBC by Metric MDS.*

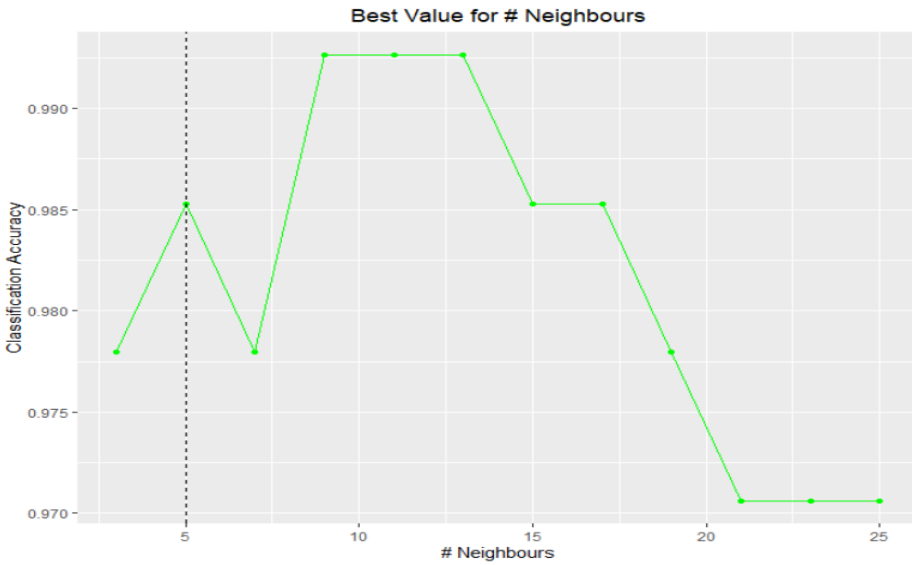

Figure A.7: *Nearest neighbour plot for reduced WDBC by ISOmap.*

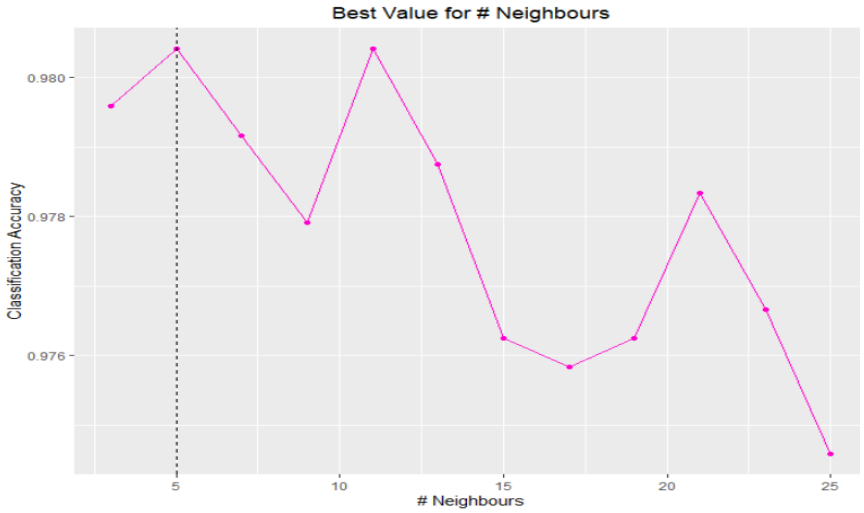

Figure A.8: *Nearest neighbour plot for reduced MNIST by PCA.*

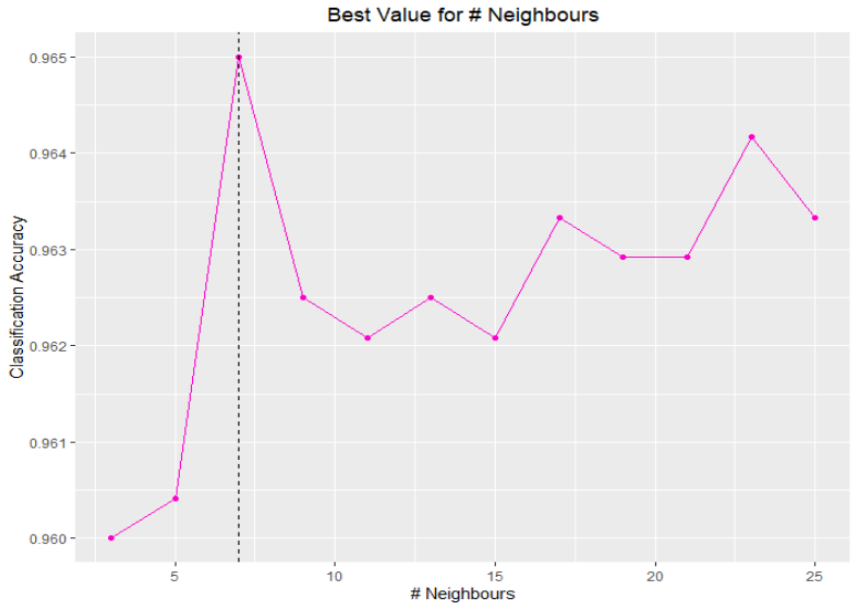

Figure A.9: *Nearest neighbour plot for reduced MNIST by Metric MDS.*

References

https://bradleyboehmke.github.io/HOML/knn.html

https://www.analyticsvidhya.com/blog/2015/08/learning-concept-knn-algorithms-programming/#:~:text=The%20knn%20()%20function%20needs,is%20a%20user%2Dspecified%20number.&text=The%20value%20for%20k%20is,of%20the%20number%20of%20observations.

https://towardsdatascience.com/principal-component-analysis-pca-101-using-r-361f4c53a9ff

https://rstudio-pubs
static.s3.amazonaws.com/246348_b31bca1e4be04bb395825dc6a00de364.html

https://rdrr.io/bioc/RDRToolbox/man/Isomap.html

https://www.rdocumentation.org/packages/RDRToolbox/versions/1.22.0/topics/Isomap

https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/Multidimensional_Scaling.pdf

https://www.rdocumentation.org/packages/RDRToolbox/versions/1.22.0/topics/Isomap

https://blog.paperspace.com/dimension-reduction-with-isomap/

Tenenbaum, J. B. and de Silva, V. and Langford, J. C. 2000. "A global geometric framework for nonlinear dimensionality reduction."