5-2023

Unsupervised Dimension Reduction Techniques for Lung Diagnosis using Radiomics

Janet Kireta
East Tennessee State University

Unsupervised Dimension Reduction Techniques for Lung Diagnosis using Radiomics

---

A thesis

presented to

the faculty of the Department of Mathematics and Statistics

East Tennessee State University

In partial fulfillment

of the requirements for the degree

Master of Science in Mathematical Sciences

---

by

Janet Akoth Kireta

May 2023

Mostafa Zahed, Ph.D., Chair

Robert Price, Ph.D.

JeanMarie Hendrickson, Ph.D.

Keywords: Radiomics, Segmentation, Dimension Reduction, Features Extraction

ABSTRACT

Unsupervised Dimension Reduction Techniques for Lung Diagnosis using Radiomics

by

Janet Akoth Kireta

Over the years, cancer has increasingly become a global health problem [12]. For successful treatment, early detection and diagnosis is critical. Radiomics is the use of CT, PET, MRI or Ultrasound imaging as input data, extracting features from image-based data, and then using machine learning for quantitative analysis and disease prediction [23, 14, 19, 1]. Feature reduction is critical as most quantitative features can have unnecessary redundant characteristics. The objective of this research is to use machine learning techniques in reducing the number of dimensions, thereby rendering the data manageable. Radiomics steps include: Imaging, segmentation, feature extraction, and analysis. For this research, a large-scale CT data for Lung cancer diagnosis collected by scholars from Medical University in China is used to illustrate the dimension reduction techniques via R, SAS, and Python softwares. The proposed reduction and analysis techniques were; PCA, Clustering, and Manifold-based algorithms. The results indicated the texture-based features as the most important in the analysis.

ACKNOWLEDGMENTS

I would like to thank my thesis supervisor, Dr. Mostafa Zahed for going way out of his way to guide, correct, recommend and generally support me through this process and making sure timely objectives were met. Also thanks to my committee members for their efforts and contributions to this work: Dr. Bob Price, and Dr. JeanMarie Hendrickson. Thanks to the East Tennessee State University Department of Mathematics and Statistics for providing financial means to complete this project and graduate school in totality. I also express my deepest gratitude to my family and friends for their unfailing support and continuous encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

The goal of studying cancer is to develop safe and effective methods to prevent, detect, diagnose, treat, and, ultimately, cure the collections of diseases we call cancer. The better we understand this disease, the more progress we will make toward diminishing the tremendous human and economic tolls of cancer. Recent advances in medical imaging, such as radiomics, have shown great potential in this regard. Radiomics allows for the extraction and analysis of large data sets from imaging techniques such as CT and PET scans. This, in turn, provides a more comprehensive understanding of tumor growth and development. As such, using radiomics in cancer detection and analysis represents a promising avenue for future research, potentially leading to significant improvements in diagnosis, treatment, and patient outcomes. The process may however turn out to be very hectic given the features obtained from radiological images are so immense. Therefore there is a dire need to have the data matrix in its simplest form to give way for prognosis, therapy, and any other objective such kinds of research would intend to accomplish. To accomplish this, the research intended to answer the following questions;

RQ 1. Is there a way we could reduce the number of variables which was 110 to a lesser number that would make the process of working with the data simple?

RQ 2. Is any of the feature categories most significant for our analysis?

Overall, the ultimate intention of the analysis would be to generate a data matrix with fewer and very significant features that can be used in the future as new predictor variables to do predictions on the Lung Cancer data.

1.1 Abbreviations

For the sake of readability, the following is a list of the main abbreviations used in this thesis:

CT	**C**omputed **T**omography
PET	**P**ositron **E**mmision **T**omography
MRI	**M**agnetic **R**esonance **I**maging
DICOM	**D**igital **I**maging and **C**ommunications in **M**edicine
XML	**E**xtensible**M**ark-up **L**anguage
PCA	**P**rincipal **C**omponent **A**nalysis
ISOMAP	**I**sometric **F**eature **M**apping
DNA	**D**eoxyribonucleic **A**cid **A**apping
cPCA	**c**ontrastive **P**rincipal **C**omponent **A**nalysis
JIVE	**J**oint and **I**ndividual **V**ariation **E**xplained
OS	**O**verall **S**urvival
PFS	**P**rogression **F**ree **S**urvival
NSCLS	**N**on **S**mall **C**ell **L**ung **C**ancer
PD-L1	**P**rogrammed **D**eath **L1**igand 1
GLCM	**G**ray **L**evel **C**o-occurence **M**atrix
GLRLM	**G**ray **L**evel **R**un **L**ength **M**atrix
GLZLM	**G**ray **L**evel **Z**one **L**ength **M**atrix
NGTDM	**N**eighborhood **G**ray **T**one **D**ifference **M**atrix
MF	**M**inkownski **F**unctional
LDA	**L**inear **D**iscriminant **A**nalysis
CCA	**C**anonical **C**orrelation **A**nalysis
NMF	**N**on-negative **M**atrix **F**actorization
FSA	**F**eature **S**election **A**lgorithm
FEA	**F**eature **E**xtraction **A**lgorithm
cmdscale	**c**lassical(metric) **m**ultidimensionalscale

Table 1: Table of abbreviations

1.2 Background

The study of cancer is of critical importance, given the global impact of this disease. While the etiology of cancer is multifaceted, the common underlying issue is

the unregulated proliferation of aberrant cells. This can lead to the development of tumors, which can be either benign or malignant. Malignant tumors are particularly concerning, as they can invade surrounding tissues and metastasize to other body parts. In 2020, cancer claimed nearly 10 million lives worldwide, making it the second most common cause of mortality. As such, the identification and diagnosis of cancer are essential for timely intervention and treatment. Medical imaging plays a vital role in these processes, allowing clinicians to probe the body's internal structures non-invasively [19]. Various modalities, such as Computed Tomography , Magnetic Resonance Imaging and Positron Emission Tomography , can be used to detect and characterize tumors. There are many different types of cancer, with some being more common than others. In 2020, lung, breast, brain, and colorectal cancers were the most common worldwide. As such, a considerable amount of research is focused on these specific types of cancer. However, studying all cancer forms is crucial to developing more effective treatments and improving patient outcomes. One emerging area of research is radiomics, which uses computational methods to extract quantitative data from medical images. This approach can provide insights into tumors' characteristics and help stratify patients according to their response to treatment. As such, radiomics has the potential to play an essential role in advancing our understanding of cancer and aiding in the development of personalized therapies.

1.3 Radiomics Overview

The field of radiomics has rapidly emerged as an important and influential area of contemporary cancer research. It offers a range of potential benefits, particu-

larly in standardizing the analysis of complex imaging data, which ultimately allows for comparative studies across multiple patients and investigations [2]. Identifying key imaging biomarkers through radiomics can significantly improve the accuracy of cancer diagnosis and staging, which can have life-saving implications for patients. Furthermore, the quantitative information that radiomics extracts from images can offer valuable insights into the underlying biology of a tumor, providing clues as to its aggressiveness or how it might respond to different treatments [16]. This information, in turn, can be used to develop tailored treatment plans for patients, identifying those most likely to benefit from specific therapies and those at a greater risk for recurrence or progression. The non-invasive nature of radiomics offers distinct advantages in reducing the need for invasive procedures and enhancing the efficiency of clinical trials [28]. Different types of non-invasive imaging include, Molecular imaging which allows clinicians to not only see where a tumor is located in the body, but also to visualize the expression and activity of specific molecules (e.g., proteases and protein kinases) and biological processes (e.g., apoptosis, angiogenesis, and metastasis) that influence tumor behavior and/or response to therapy, Anatomical imaging enables detection of a phenotypic(physical expression of DNA(Deoxyribonucleic acid)) alteration that is sometimes, but not invariably, associated with cancer and finally, Functional imaging used to study tumor physiology, to probe tumor molecular processes, and to study tumor molecules and metabolites in vitro and in vivo. These attributes make radiomics an exciting and promising field poised to contribute significantly to advancing cancer research and treatment.

Radiomics often encompasses the extraction and analysis of quantitative features

from medical images, including but not limited to CT and PET scans. By evaluating tumor size, shape, texture, and density, radiomics offers a promising avenue for advancing personalized medicine [1]. CT and PET scans are widely employed medical imaging techniques that play an essential role in diagnosing and monitoring cancer. While similar in that they are both non-invasive, the two methods differ in how they generate images. CT scans use x-rays to create detailed, cross-sectional images of internal organs and structures, which can help doctors identify the location and size of tumors. On the other hand, PET scans involve injecting a small amount of radioactive material into the body, which is then used to produce images that reveal the functional activity of tissues (John Hopkins Medicine, 2021). Doctors can analyze these images to assess how cancer cells metabolize nutrients, grow, and spread. Together, these two imaging techniques provide a comprehensive way to monitor cancer without requiring invasive procedures.

One critical step in the radiomics workflow is feature extraction, which involves identifying and quantifying the various characteristics of tumors. To accomplish this, segmentation is typically performed to isolate the tumor region, and then multiple methods are used to extract features based on tumor intensity, texture, and shape [29]. Dimension reduction techniques, such as PCA and clustering, are often used to help process and analyze these features. These techniques help to simplify the data by reducing the number of variables and identifying key patterns. More advanced methods have been developed for dimension reduction, such as contrastive Principal Component Analysis (cPCA) and Joint and Individual Variation Explained (JIVE). The cPCA approach can identify low-dimensional structures unique to a particular

data set by comparing them to a reference data set. JIVE, on the other hand, decomposes variation across multiple data types into joint and individual components [21]. Both methods can help analyze complex medical imaging data. Some of the software tools used for feature extraction include PyRadiomics, 3D Slicer, LIFEx, IBEX, QIFE, and RayPlus. Each device has strengths and limitations, so researchers must carefully consider which best meets their needs.

Overall, the implications of radiomics as a field of study are substantial, particularly as they pertain to diagnosing, treating, and monitoring cancer. By utilizing quantitative data extraction methods from medical images, radiomics can allow researchers to discern patterns in tumor biology that might otherwise remain obscured. This may help shed light on various aspects of a tumor's behavior, such as its aggressiveness or responsiveness to different treatment modalities. As such, radiomics has the potential to contribute significantly to our overall understanding of cancer and to facilitate the development of more effective and personalized therapies.

2 LITERATURE REVIEW

The use of radiomics in cancer detection and analysis is an area of research that has continued to grow in recent years. Many studies have demonstrated the potential benefits of using this approach in various types of cancer. [5] highlights that radiomics can be employed in multiple ways to support the management of lung cancer, such as by aiding in the detection and classification of pulmonary nodules, assessing histopathology and genetics, staging the disease, and predicting response to therapy and prognosis. The ability of radiomics to extract quantitative features from medical images and identify associations with clinicopathological information can provide valuable insights that are not readily apparent through conventional analysis. Similarly, a study by [29] contributes to the growing body of literature exploring the potential of radiomics in lung cancer diagnosis and prognosis.

By examining the performance of eight selected radiomic features in three independent cohorts, the authors demonstrate these features' value for predicting long-term prognosis and characterizing tumor heterogeneity. In particular, the feature "kurtosis" emerged as a promising metric for lung cancer classification and progression. However, the authors emphasize combining multiple radiomic features to achieve better prognostic accuracy. Overall, the study offers valuable insights into the clinical application of radiomics in lung cancer while highlighting areas for further research.[11] present a comprehensive analysis of the potential for radiomics to predict Overall Survival (OS) and Progression-Free Survival (PFS) in patients with Non-Small-Cell Lung Cancer (NSCLC). While the authors highlight that several studies have demonstrated the predictive value of Radiomics Features (RFs) in NSCLC, they also point

out that no radiomics-based model is currently clinically validated or used in routine practice. In their study, [11] found that while some RFs had prognostic value, adding radiomics data to conventional data did not improve survival prediction. Additionally, they did not find any association between PD-L1 (Programmed death-Ligand 1) expression and RFs.

The article by [8] provides a comprehensive overview of the use of radiomics in lung cancer detection, diagnosis, and treatment. The authors highlight the development of radiologic features from semantic and handcrafted radiomics to deep radiomics features, summarizing the latest applications of structural and functional radiomics on early and advanced-stage lung cancer. While the article acknowledges some limitations and challenges associated with radiomics, such as the need for extensive and diverse datasets, reproducibility concerns, and the lack of standardized methodologies, the authors point to future directions for research, including the potential for federated learning and multidisciplinary convergence. Overall, the article provides a valuable contribution to the field, offering a current and forward-looking perspective on applying radiomics in lung cancer. These studies, among others, demonstrate the potential of radiomics in the detection and analysis of cancer. By providing a deeper understanding of tumor biology and behavior, radiomics can improve diagnosis, prognosis, and treatment outcomes.

3 RESEARCH METHODOLOGY

The research, therefore, explored the techniques used to address the research question. These include data description and analysis techniques.

3.1 Data description

The data was collected by Huiping Han, Funing Yang, and Rui Wang of Harbin from the Medical University in Harbin in China [29]. This data is available on The Cancer Imaging Archive (TCIA). The workflow of radiomics includes; Medical imaging followed by segmentation performed to define the tumor region. From this region, the features are extracted based on tumor intensity, texture, and shape[18]. Finally, these features are used for analysis, Figure 1.

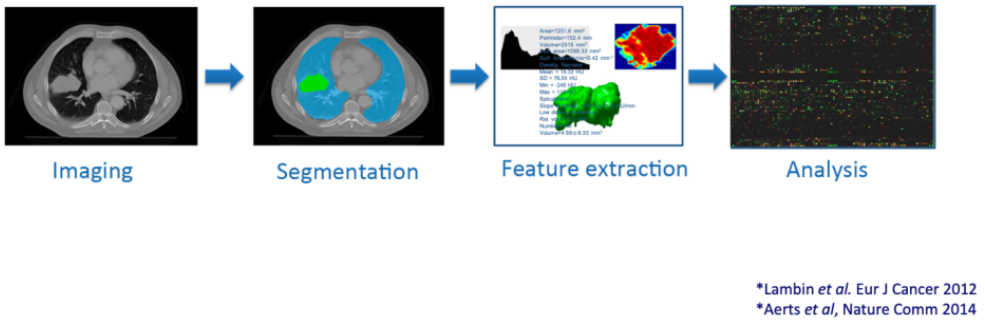


Figure 1: The radiomics workflow.

This dataset consists of CT DICOM images of 130 patients with lung cancer. The XML Annotation files which include the location of the tumor were provided by five academic radiologists with high expertise in lung cancer. To visualize the annotation boxes on the tumor of the DICOM images[18], python codes through the terminal

were used to pull out the images and put them in a box, Figure 2.

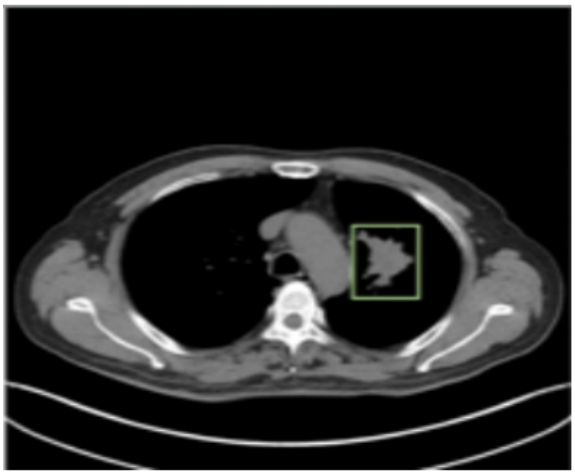


Figure 2: Visualization of the annotation box on the CT-DICOM images.

Source: https://wiki.cancerimagingarchive.net

3.1.1 The process of Data Acquisition

a) **Software tools for extracting features.**

There are massive software tools available for extracting tumor features from medical images. Some standard options include PyRadiomics, 3D Slicer, LIFEx, IBEX, QIFE, and RayPlus. Each of the devices has its drawbacks and advantages. It is therefore at the researcher's discretion to identify which best aligns with his intended objectives. For example, PyRadiomics is a flexible open-source platform capable of extracting a wide array of features, but it requires some programming knowledge in Python [9]. 3D Slicer, on the other hand, is a free and open-source application designed to facilitate the development of new functionality in 3D Slicer extensions [7], Figure 3. LIFEx is another option that

offers a user-friendly interface and powerful features for tumor segmentation, feature extraction, and radiomics analysis. Ultimately, the choice of software tool depends on the researcher's goals and expertise,

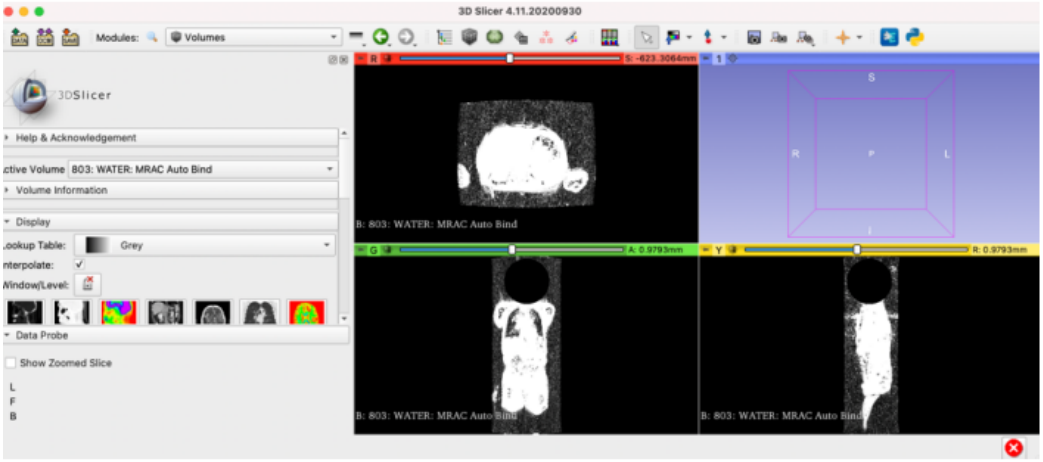
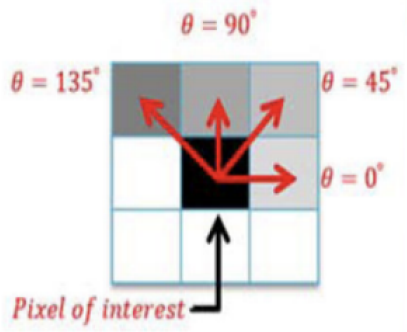


Figure 3: Loading Lung-CT-PET images.

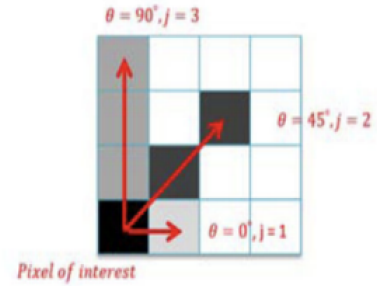
Source: https://wiki.cancerimagingarchive.net

b) **Extracting features from CT medical images of lung cancer.**

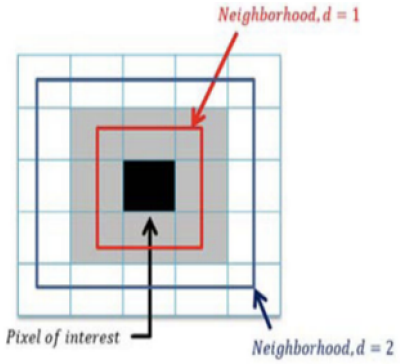
Features that are extracted can be generally classified into three main categories [2]: **First order radiomics** which has Intensity-based features and Shape-based features, **second order radiomics** which has Texture-based features extracted based on different descriptive matrices (Gray level co-occurrence matrix (GLCM), Gray level run length matrix (GLRLM), Neighborhood gray-tone difference matrix (NGTDM), Gray level zone length matrix (GLZLM), Figure 4.



(a) GLCM (gray level co-occurence matrix).



(b) GLRLM (gray level run length matrix).



(c) NGTDM (neighborhood gray-tone difference matrix).

Figure 4: Texture Features Parekh and Jacobs (2016).

The last category, **higher order radiomics** applies the use of filters to extract features from images through wavelet which decomposes tumor images into different frequency domains (such as horizontal, vertical, and diagonal) and then extracts the tumor shape, intensity, texture, and other information. Fourier

features capture gradient information while Minkowski Functional (MF) is a common higher-order feature extractor considering the patterns of pixels with intensities above a predefined threshold.

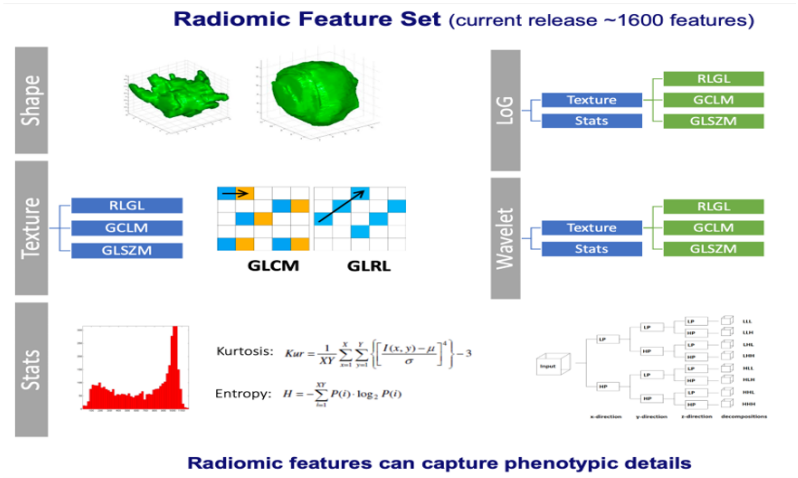


Figure 5: Categories of features.

Source: https://wiki.cancerimagingarchive.net

c) **Extraction process:** Out of the 130 patients under consideration, the extraction of features was done on 74 patients because the provided annotation files did not work for all 130 patients. A 3D slicer was used to do the segmentation process as indicated by the yellow circle around the blue and pink colors on the tumor, Figure 6.

The PyRadiomics package available in the 3D slicer was then used to extract features from the tumor segmentations for all patients, Figure 7.

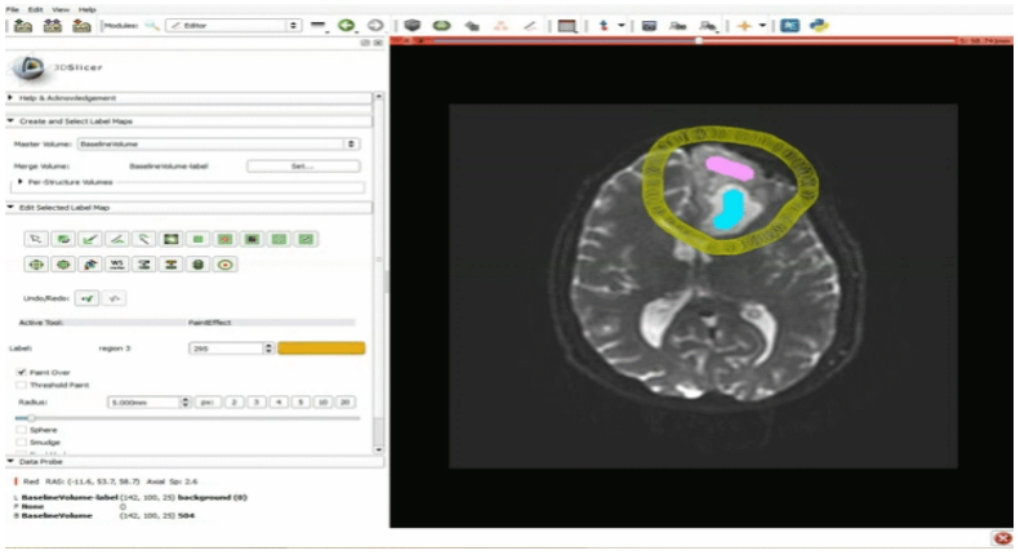


Figure 6: A 3D slicer segmenting the tumor.

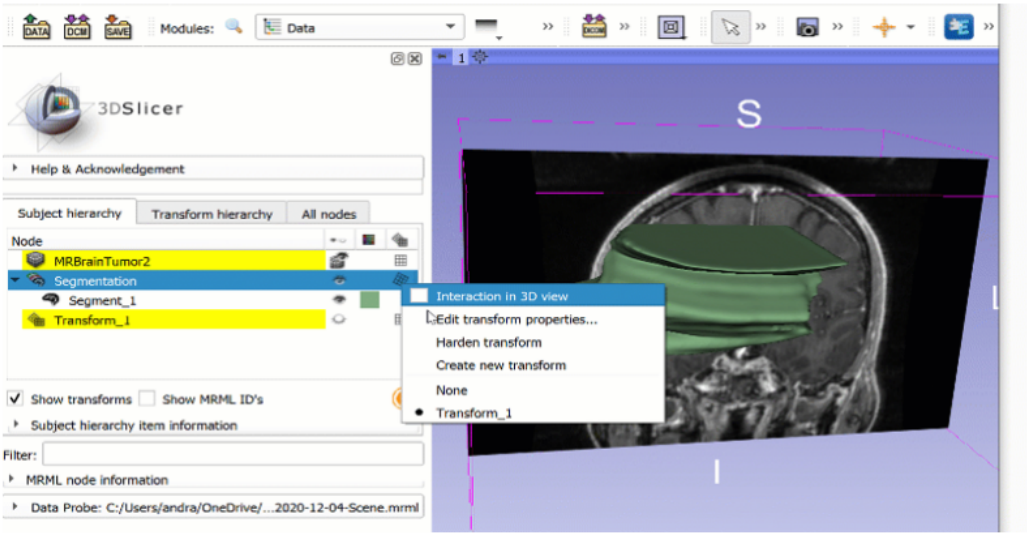


Figure 7: Pyradiomics package extracting features.

d) **Resulting Data Matrix** After the whole process of extraction, an unsupervised data matrix was obtained with a dimension of 74 by 110. Each row represented the patient, and each column was for the extracted feature depending on the categories earlier discussed. The 110 features acquired were quantitative variables. Finally, the data matrix was normalized according to the min-max normalization approach as it is robust to any feature distributions and leads to making unitless measurements for each feature[18]. The features on the columns were renamed since the original names were too long to enable data visualization through graphs. The column names range from *diagnostic_Image.original_Maxim,..., original_ngtdm_Strength_CT* were renamed to $V_1, \ldots, V_{10}$. Since the resulting matrix has 74 rows by 110 columns, most reduction techniques algorithms such as PCA, hierarchical and even k-means clustering can not handle such a data format successfully, it is for this reason that the normalized data set was transformed into a square correlation matrix such that the new dimension was 106 by 106. It is after this transformation the data matrix was finally ready for applying dimension reduction techniques.

3.2 Data Analyses

As directed by the research objective, dimension-reduction techniques were applied to render the data more manageable. These approaches included feature extraction and selection [27]. Feature extraction techniques are further categorized into; supervised and unsupervised learning. Supervised learning is a technique that considers the relation of features with class labels and features are selected mostly based on

their contribution to distinguish classes, while, unsupervised learning does not consider the class labels and its objective is to remove redundant features[4]. Because the obtained data matrix is unsupervised, therefore a further exploration into the classification of unsupervised learning techniques whether linear (Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Canonical Correlation Analysis (CCA)) or non-linear (e non-negative Matrix Factorization (NMF) and manifold-based methods) was done to transform the original data into a new set of features that retain most of the original dataset's information. Equally, feature selection methods aim to discern the original dataset's most relevant or informative features. Methods of feature selection include filter methods (e.g. low variance filter selection, information gain, chi-square test, Fisher's score), wrapper methods (e.g. forward selection, backward elimination, exhaustive feature selection), and embedded methods (e.g. regularization, random forest importance). Selecting the appropriate dimension reduction technique is a function of the specific dataset and research objectives. Employing these techniques allows researchers to improve computational efficiency, avoid the curse of dimensionality, and pinpoint the most salient features in the dataset, Figure 8.

3.2.1 Feature Extraction Algorithm (FEA)

The objective of feature extraction algorithms is to convert unprocessed data into a collection of features that more accurately reflect the intrinsic patterns present within the data [10]. Although feature extraction algorithms can be implemented in various ways, they generally fall into two primary categories: linear and non-linear.

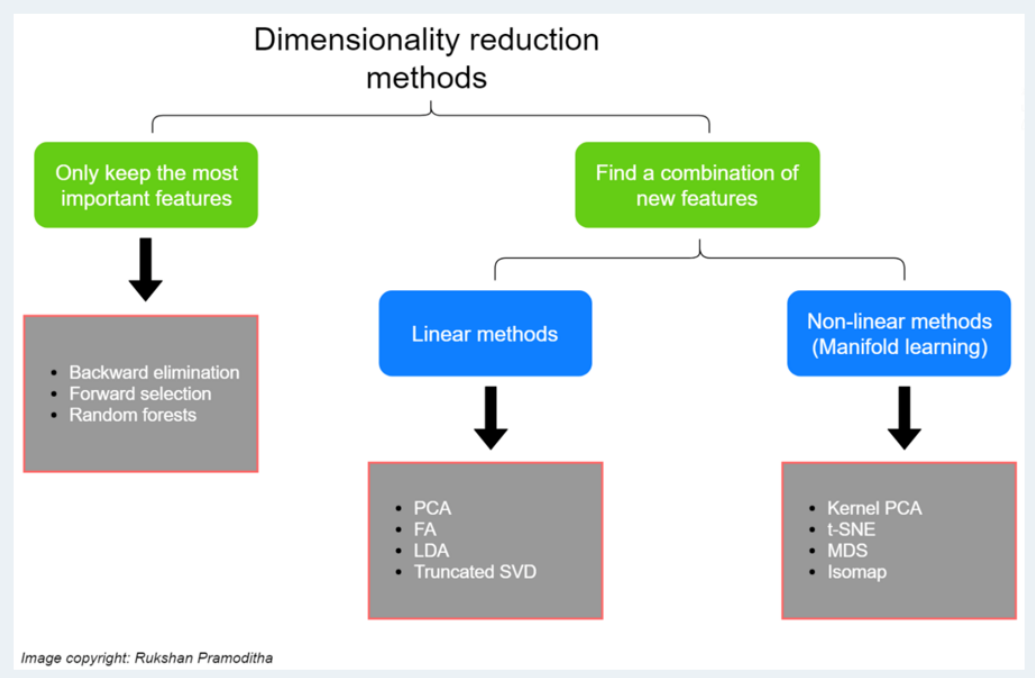


Figure 8: Dimensionality reduction techniques.

i) Linear Analysis is used when features have linear correlations. The proposed linear analysis most ideal for this study was Principal Component Analysis (PCA), which is a feature transformation technique that reduces the correlation between sampled variables [15] say $x_1, x_2, ..., x_p$. Using an orthogonal transformation, PCA generates new variables referred to as principal components $PC_1, PC_2, ...PC_m$ that retains many of the properties of the original variables given $m < p$. This approach enables the creation of various features through linear combinations of the main components, which maximize variance and improve predictability [6], Figure 9.

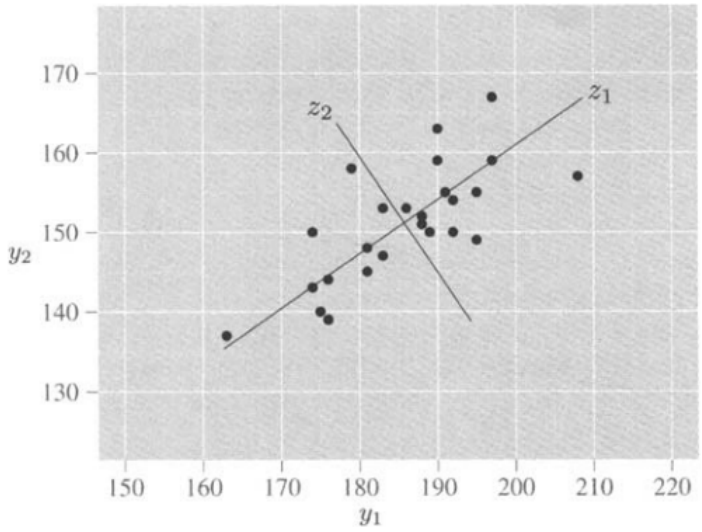


Figure 9: Principal component analysis.

Source: Rencher and Christensen (2012)

There are five main steps to conducting PCA:

(a) **Standardize the data:** Calculate the mean of all the dimensions of the data set, except the labels. Scale the data so that each variable contributes equally to the analysis. In the equation given below, z is the scaled value, x is the initial, and μ and σ are the mean and standard deviation, respectively.

$$Z = \frac{x - \mu}{\sigma}$$

(b) **Compute the covariance matrix:** Identifying highly correlated variables is a crucial step in data analysis. These variables often contain redundant information, which can hinder the accuracy of statistical models and analyses. Utilizing a covariance matrix allows for the examination of

correlations between all possible variable pairs within a given data set and facilitates the removal of any superfluous variables.

$$cov(x, y) = \frac{1}{n-1}\Sigma_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y}),$$

where $\overline{x}$ is the mean of the predictor variables, $\overline{y}$ is the mean of the response variables, n is the sample size and i refers to each observation.

Basing the PCA on the covariance matrix would however lead to variables with large variances dominating the most important principal components. Also, changing the units of measurement (e.g., from ounces to pounds, or from feet to inches) would change the PCA solution. For this reason, it is often preferred to base the PCA solution on the eigenvectors and eigenvalues of the correlation matrix rather than the covariance matrix. This is equivalent to initially standardizing all variables and then performing the PCA based on correlation matrix[22].

(c) **Calculate the eigenvectors and eigenvalues:** Using concepts originating from linear algebra enables determining principal components stemming from the covariance matrix. An eigenvalue is a scalar that is used to transform (stretch) an eigenvector. The relevant equation is as follows:

$$Av = \lambda v,$$

where A is the square covariance matrix, v is an eigenvector, λ is a scalar which is eigenvalue (associated with eigenvector of A matrix.A solution of this equation would yield λ eigenvalue:

$$det(A - \lambda I) = 0,$$

where *det* is the determinant, A is the covariance matrix, λI is a scalar multiplying an identity matrix.

(d) **Choose k eigenvectors with the largest eigenvalues:** Sort the eigenvalues corresponding to eigenvalues from highest to lowest. In case the goal is to decrease the dimension to two from three, take the first two eigenvectors which are corresponding to the first two highest eigenvalues.

(e) **Remodel the data:** The final step uses the information from the eigenvectors of the covariance matrix to reorient data from the original axes to the ones that are now represented by the principal components.

$$y = W^{\top} x,$$

where $W^{\top}$ is the transpose of the matrix W, X is the eigenvector matrix and y is the transformed data set

Assuming our set of variables in the original data is $x_1, x_2, ...x_p$ after transformation the first principal component will be $z_1 = a_{11}x_1 + a_{21}x_2 + ... + a_{p1}x_p$,where $a_{11}, a_{21}, ..., a_{p1}$ are the loadings of the first principal component. As earlier illustrated the loadings are values of the eigenvector of the covariance matrix. Since the eigenvalues are variances of the principal components, we can speak of "the proportion of variance explained" by the first k components[18]: proportion of variance $\frac{\lambda_1+\lambda_2+,...,\lambda_k}{\lambda_1+\lambda_2+,...,\lambda_p}$, where λ_1 refers to the variation explains by the first component, and so on [22].

To be able to come up with these principal components according to [22];

(a) We can retain the first m components sufficient to explain a specified percentage (70% 80% 90% of the total variance of the original variables).

(b) Keep components whose eigenvalues are at least $\Sigma\frac{\lambda_i}{p}$ which is the average eigenvalue and also the average sample variance of the original variables, where λ_i, is a constant λ multiplying the number of factors i and p is the total number of observations in the data set.

(c) Use a Scree plot which is a plot of the eigenvalues λ_i, where λ is a constant and i is the number of factors. It always displays a downward curve. The point where the slope of the curve is leveling off (the "elbow) indicates the number of factors that should be generated by the analysis as, Figure 10.

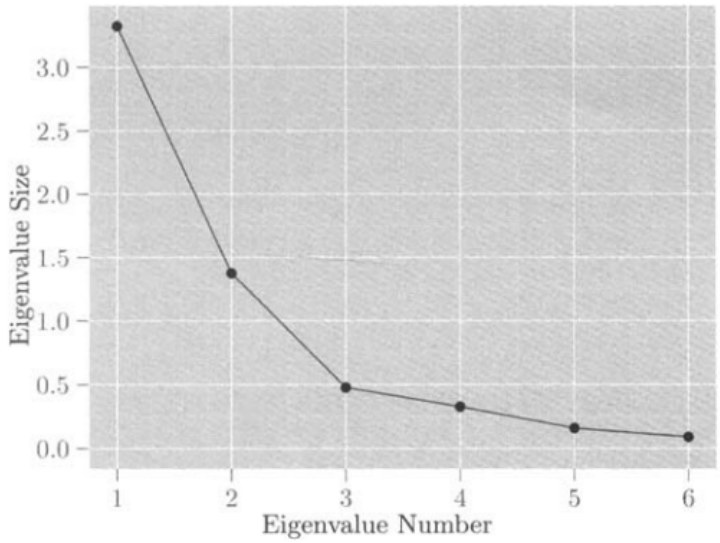


Figure 10: Scree plot.

Source: Rencher and Christensen (2012)

ii) Non-Linear Analysis is used when features are assumed not to lie on a linear

space. For this study, manifold-based algorithms which is an approach that executes tasks by the use of algorithms to project data into a lower dimensional space were chosen. In line with the resolve that visualization of high-dimensional data sets can be cumbersome and less intuitive, there is a need to reduce the dimensions so that a few remaining dimensions can be plotted. The easiest way to achieve this is by a random projection of the data, however, interesting structures within the data may be lost. Steps towards achieving this objective involve;

- Building a neighborhood graph from the given data.
- Computing the shortest-path distances along the graph.
- Applying multidimensional scaling to find a low-dimensional representation.

(a) Isometric Feature Mapping (ISOMAP): It is an algorithm that projects data to a lower dimension space while preserving distances between data points retaining the geodesic distance rather than the Euclidean distance. This allows it to capture the nonlinear structure of the data more effectively. The geodesic distance of two data points that live in a manifold is the shortest distance along the manifold, on a sphere, in other words, the great circle distance[25], Figure 11

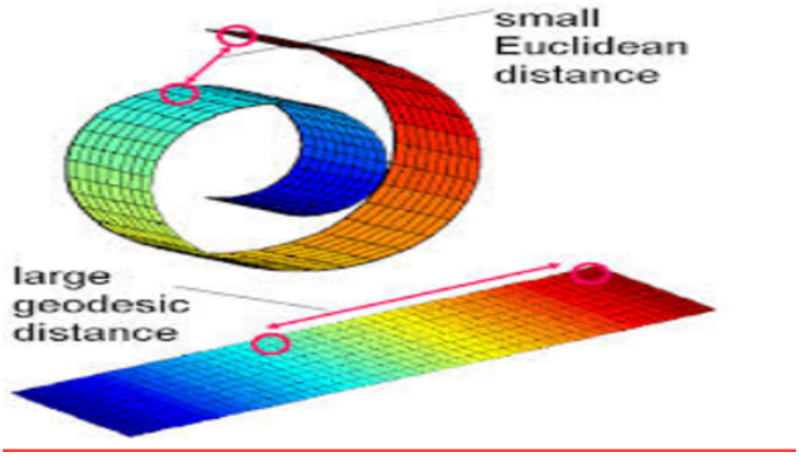


Figure 11: A 3D swissroll showing euclidean and geodisic distance.

Source: https://web.mit.edu/cocosci/isomap/isomap.html

The ISOMAP Algorithm steps include;

- Construct a neighborhood graph G from the given data by connecting only "nearby points" with edges weighted by their Euclidean distances, i.e., $d_X(i,j) = ||x_i - x_j||$ if x_i, x_j are "close" (and 0 otherwise) where "closeness" is defined in one of the following ways:

  i) ϵ - ball approach: For each x_i another point x_j is close if and only if $x_i - x_j < \epsilon$, Figure 12, or

  ii) kNN approach: For each point x_i, x_j is close if it is among the k nearest, Figure 12.

- Compute the shortest-path distances by applying Dijkstra's algorithm 1 with the nearest neighbor graph G (constructed by either method) to find the shortest-path distances for all pairs of data points ($d_G(i,j)$),

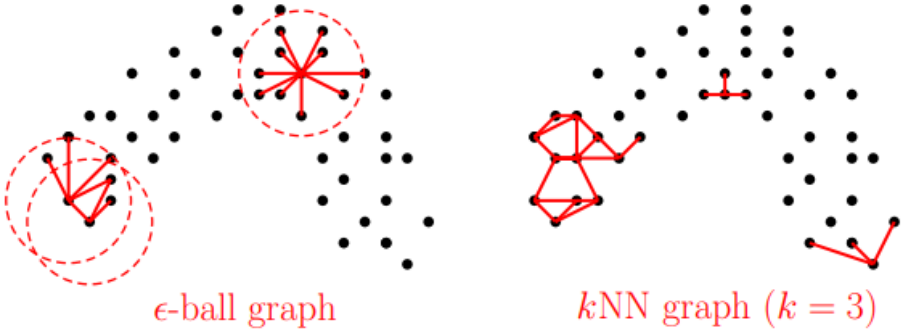
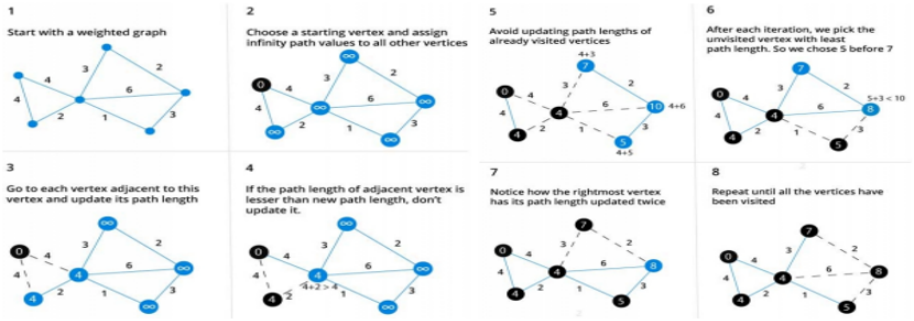


Figure 12: Epsilon ball and KNN graphs.

Source: https://web.mit.edu/cocosci/isomap/isomap.html

Figure 13.



(Source: https://www.programiz.com/dsa/dijkstra-algorithm)

Figure 13: Dijkstra's algorithm1.

- Apply MDS with $d_G(i, j)$ as input distances to find a k -dimensional representation Y of the original data.

(b) Laplacian Eigen Maps of Data Mapping It employs a geometrically-motivated algorithm for non-linear dimensionality reduction and relies on spectral techniques to project data into a lower dimensional space. It aims to preserve locality; that is, it endeavors to ensure that data points close to one another in the high dimensional space remain so in the lower dimensional space. Ultimately, the method calculates the weights which are most successful in reconstructing the vectors from their neighbors before generating the low-dimensional vectors which are best reconstructed by these weights[3].

The Laplacian Eigenmaps algorithm;

Input: $x_1, \ldots, x_n \epsilon \mathbb{R}^d$, embedding dimension $k \geq 1$, neighborhood graph method (ϵ -ball or kNN), weighting method (binary or Gaussian).

Output: A k-dimensional representation of the input data ($y \epsilon \mathbb{R}^{n*k}$).

Steps:

- Construct a neighborhood graph G from the given data.
- Set the edge weights using the specified method to form the weight matrix W.
- Compute the normalized graph Laplacian

  $L_{rw} = D^{-1}L = D^{-1}(D - W) = 1 - D^{-1}W$, where D=diag(W_1).
- Find the eigenvectors of L_{rw} corresponding to the second to $(k+1)^{st}$ smallest eigenvalues

  $L_{rw}v_i = \lambda_i v_i$, where $i = 2, \ldots, k+1$
- Return: $Y = [v_2\, ldots, v_{k+1}] \epsilon \mathbb{R}^{n*k}$.

Clustering Analysis : It separates individual observations into groups based on the values for the p variables measured on each individual.

a) Hierarchical Clustering

Agglomerative hierarchical clustering begins with n clusters, each containing a single object. At each stage, the two clusters that are "closest" are merged. As the stages iterate, there are n clusters, then n-1, and so on. By the last stage, there is 1 cluster containing all n objects, Figure 14.

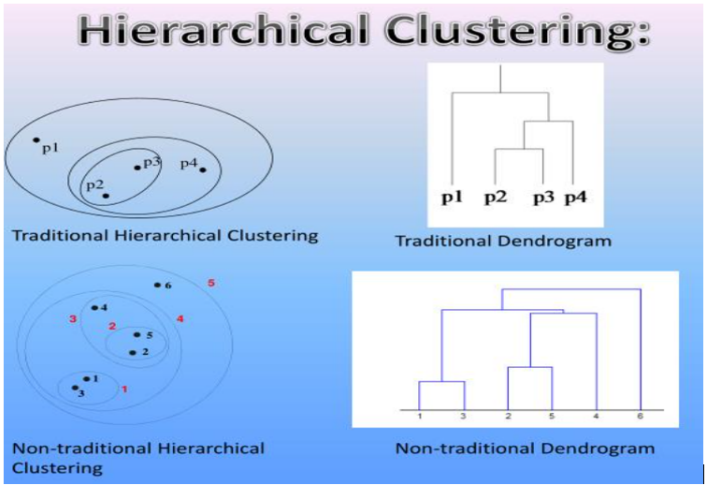


Figure 14: Hierarchical clustering.

Source: https://www.slideserve.com/lenci/partitional-clustering

There are four common types of linkage: complete, average, single (Ward's), and centroid. The summary of these linkages is as follows [13][18]:

- **Complete:** In this approach, all pairwise dissimilarities between the observations in the clusters are computed and the maximum one will be

recorded[18].

- **Single (Ward's):** In this method, all pairwise dissimilarities between the clusters are computed and the minimum one will be recorded[18].
- **Average:**In this approach, all pairwise dissimilarities between the clusters are computed and the average of dissimilarities will be recorded.
- **Centroid:** In this technique, the dissimilarities between the mean vector of for cluster. A (centroid) and the mean vector for cluster B (centroid) are computed[18].

Below are the steps in the agglomerative hierarchical clustering algorithm for grouping N objects according to [22]:

i) Start with N clusters, each containing a single entity and an N x N symmetric matrix of distances (or similarities) $D = d_{ik}$

ii) Search the distance matrix for the nearest (most similar) pair of clusters. Let the distance between "most similar" clusters U and V be d_{uv}

iii) Merge clusters U and V. Label the newly formed cluster (UV). Update the entries in the distance matrix by,

(i) deleting the rows and columns corresponding to clusters U and V and

(ii) adding a row and column gives the distances between cluster (UV) and the remaining clusters.

iv) Repeat steps 2 and 3 a total of $N-1$ times. (All objects will be in a single cluster after the algorithm terminates.) Record the identity of clusters that

are merged and the levels (distances or similarities) at which the mergers take place.

b) k-means

The k-means algorithm [17] begins by randomly allocating the n objects into k clusters (or randomly specifying k centroids). One at a time, the algorithm moves each object to the cluster whose centroid is closest to it, using the measure of closeness. When an object is moved, the centroids are immediately recalculated for the cluster gaining the object and the clutter losing it. The method repeatedly cycles through the objects until no reassignments of objects take place. The final clustering result will somewhat depend on the initial configuration of the objects.

The k-means clustering results from a fundamental mathematical idea; Assume that $C_1, C_2, ..., C_k$ represents sets including the observations clustered into K subgroups of the original data. These sets meet two properties[13][18]

- $C_1 U C_2 U ... U C_k = 1, ..., n$. It means the union of all clusters leads to the whole observation[18].
- $C_k n {C_k}^{\top} = \phi$ for all $k \neq k^{\top}$. It means clusters are pairwise and mutually exclusive[18].

The algorithm behind k-means clustering techniques[18] includes;

i) Randomly assign a number to each observation from 1 to K. This calls for an initial clustering of the observations[18].

ii) Repeat the following process till the cluster assignments stop changing[18].

- For each of the K clusters, calculate the k^{th} cluster centroid which is the vector of the p feature means for the observations in the k^{th} cluster[18].
- Use Euclidean distance for assigning each observation to the nearest centroid[18].

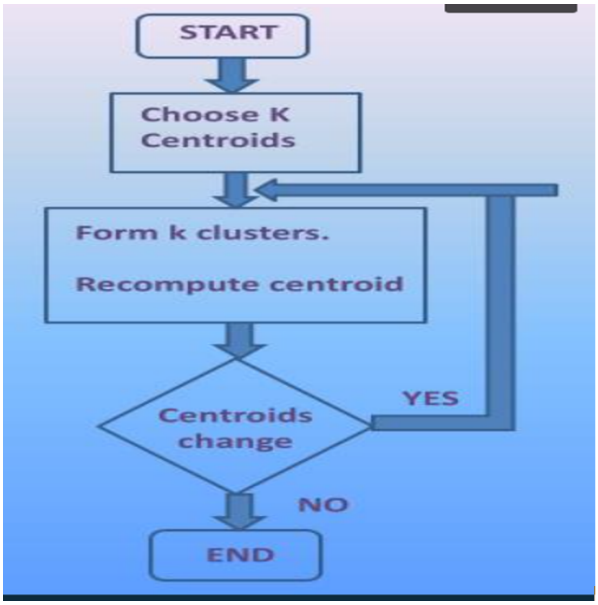


Figure 15: K-means clustering.

Source: https://www.slideserve.com/lenci/partitional-clustering

3.2.2 Feature Selection Algorithm (FSA)

Feature selection is selecting the most relevant structures from a dataset to build an accurate and efficient model. There are three general approaches to feature selection: filter, wrapper, and embedded methods. Filter methods assess the utility of individual features by quantifying their statistical relationship with

the target variable. Various filter methods exist, but for our research, we will focus on variance and loadings which can be used alongside other reduction techniques to augment the process of variable selection. The filters can be used to identify the features with little or high variability across the samples. Low variance features can be considered less informative or redundant while high variance would bring a lot of dissimilarities more so in clustering analysis, and removing them can simplify the analysis and potentially improve its accuracy or complicate the data introducing errors.

The process involving using the filter as a dimension reduction technique is to set threshold values based on calculations such as five-number summary and remove any features below or above such values. It is however necessary to carefully select the appropriate threshold and consider the potential trade-offs between dimensionality reduction and information loss.

4 SUMMARY OF FINDINGS, CONCLUSIONS, AND RECOMMENDATIONS

4.1 Principal Component Analysis

With the data set on 110 features extracted from tumors of CT images of lung cancer patients, an illustration of how the reduction technique were used was shown. This particular analysis entailed both the use of SAS and R soft wares consecutively. The data was standardized through SAS such that each variable had a mean of zero and a standard deviation of one. A partial table of features before and after standardization is, Table 2.

Before Standardization			After Standardization	
Variable	Mean	Std Dev	Mean	Std Dev
diagnosticsImage.originalMaximumCT	2795.041	561.835	0.00	1.00
diagnosticsMask.originalVoxelN	1983.781	1983.918	0.00	1.00
diagnosticsMask.originalVolume	1.973	2.387	-0.00	1.00
originalshapeElongationCT	0.675	0.173	0.00	1.00
originalshapeFlatnessCT	0.003	0.0025	-0.00	1.00
originalshapeLeastAxisLengthC	0.363	3.119	-0.00	1.00
originalshapeMajorAxisLengthC	47.262	24.575	0.00	1.00
originalshapeMaximum2DDiameter	39.061	22.230	0.00	1.00
originalshapeMaximum2DDiameterRowCT	41.043	23.971	0.00	1.00
originalshapeMaximum2DDiameterSliceCT	53.315	27.032	-0.00	1.00
originalshapeMaximum3DDiameter	54.407	28.796	-0.00	1.00
originalshapeMeshVolumeCT	834.308	827.157	0.00	1.00
originalshapeMinorAxisLengthCT	31.080	16.408	0.00	1.00
originalshapeSphericityCT	0.213	0.078	0.00	1.00
originalshapeSurfaceAreaCT	2492.290	2419.104	-0.00	1.00
originalshapeSurfaceVolumeRatioCT	3.135	0.280	-0.00	1.00
originalshapeVoxelVolumeCT	857.009	836.936	0.00	1.00
originalfirstorder10PercentileCT	-286.601	211.378	0.00	1.00
originalfirstorder90PercentileCT	137.711	101.039	0.00	1.00
originalfirstorderEnergyCT	94411905.230	101889295.88	-0.00	1.00
originalfirstorderEntropyCT	4.505	0.547	-0.00	1.00
originalfirstorderInterquartileRangeCT	191.916	106.791	0.00	1.00

Table 2: A partial table of the means and standard deviations of some of the features before and after standardizing.

Principal components are computed from the correlation matrix, so the total variance is equal to the number of variables which is 110, Figure 16.

The PRINCOMP Procedure

Observations	73
Variables	110

	diagnostics_Image.original_Maxim	diagnostics_Mask.original_VoxelN	diagnostics_Mask.original_Volume	original_shape_Elongation_CT	original_shape_Flatness_CT	original_shape_LeastAxisLength_C	origi
Mean	2847.356164	1983.780822	1.931506849	0.6751253252	0.0029761177	0.367544551	
StD	338.669265	1953.917527	2.376556608	0.1732133102	0.0254279605	3.140302018	

Figure 16: Number of observations and simple statistics.

SAS software computes the principal components from the correlation matrix, a partial representation of the correlation matrix was shown, Figure 17.

	diagnostics_Image.original_Maxim	diagnostics_Mask.original_VoxelN	diagnostics_Mask.original_Volume	original_shape_Elongation_CT	original_shape_Flatness_CT
diagnostics_Image.original_Maxim	1.0000	0.1846	-.0860	-.1765	0.0241
diagnostics_Mask.original_VoxelN	0.1846	1.0000	-.2099	0.1596	-.0370
diagnostics_Mask.original_Volume	-.0860	-.2099	1.0000	-.2073	0.7524
original_shape_Elongation_CT	-.1765	0.1596	-.2073	1.0000	-.2047
original_shape_Flatness_CT	0.0241	-.0370	0.7524	-.2047	1.0000
original_shape_LeastAxisLength_C	0.0241	-.0370	0.7524	-.2047	1.0000
original_shape_MajorAxisLength_C	0.1511	0.8232	0.1378	-.2081	0.3654
original_shape_Maximum2DDiameter	0.1600	0.7468	0.2643	0.0052	0.5101
VAR9	0.1214	0.8360	0.1460	0.0323	0.4022
VAR10	-.0085	0.7988	0.0413	-.1395	0.0202
original_shape_Maximum3DDiameter	-.0001	0.7378	0.2842	-.1978	0.3452
original_shape_MeshVolume_CT	0.1859	0.9977	-.2168	0.1616	-.0457
original_shape_MinorAxisLength_C	0.1410	0.9465	-.0743	0.2835	0.1107
original_shape_Sphericity_CT	-.0496	-.7864	0.1177	-.1742	-.0518
original_shape_SurfaceArea_CT	0.1817	0.9983	-.2094	0.1610	-.0372
original_shape_SurfaceVolumeRati	-.0089	-.5269	0.2902	-.2287	0.0971
original_shape_VoxelVolume_CT	0.1850	0.9978	-.2104	0.1591	-.0384
original_firstorder_10Percentile	0.0596	0.1665	-.0920	0.1858	0.0726
original_firstorder_90Percentile	0.5016	0.2827	-.2348	-.0525	0.0795
original_firstorder_Energy_CT	0.1033	0.7841	-.1025	0.0638	-.0393
original_firstorder_Entropy_CT	0.4325	0.1696	0.0206	-.1131	0.0306
original_firstorder_Interquartil	0.2097	-.1255	0.0403	-.2168	-.0101
original_firstorder_Kurtosis_CT	-.1055	0.1668	0.0177	0.0086	0.0790

Figure 17: Correlation matrix.

R software was then used for the remaining part of the analysis to generate some desired results like the scree plot among others. By using eigen values as a way of selecting principal components using, a summary table informed the conclusion that the first principal component accounted for about 59.3% of the total variance, the second principal component accounted for about 29.0%, and the third principal

component accounted for about 5.1%. Note that the eigenvalues sum to the total variance. The eigenvalues indicated that three components provide a good summary of the data accounting for 93% of the total variance while the rest of the components only account for less than 4% each, Table 2.

Eigenvalue	**Variance.percent**	**Cumulative.**	**variance.percent**
Dim.1	12.11374	59.37771	59.37771
Dim.2	5.92293	29.03233	88.41004
Dim.3	1.037131	5.083689	93.49373
Dim.4	0.781042	3.828422	97.32215
Dim.5	0.249191	1.221458	98.5436
Dim.6	0.097194	0.476414	99.02002
Dim.7	0.076179	0.373403	99.39342
Dim.8	0.05263	0.257974	99.6514
Dim.9	0.034513	0.169171	99.82057
Dim.10	0.012128	0.059446	99.88001
Dim.11	0.007384	0.036196	99.91621
Dim.12	0.006432	0.031528	99.94774
Dim.13	0.004955	0.024286	99.97202
Dim.14	0.00153	0.0075	99.97952
Dim.15	0.000907	0.004445	99.98397
Dim.16	0.000797	0.003907	99.98788
Dim.17	0.000737	0.003613	99.99149
Dim.18	0.000444	0.002177	99.99367
Dim.19	0.000349	0.001713	99.99538
Dim.20	0.000254	0.001245	99.99662
Dim.21	0.000168	0.000822	99.99745
Dim.22	0.000143	0.000699	99.99815

Table 3: Principal component analysis of the first 22 features.

A graphical representation of how many principal components should be retained to summarize our data was used. The graph is a scree plot which is a plot of the eigenvalues λ_i against factor i. It always displays a downward curve. The point where the slope of the curve is clearly leveling off (the "elbow) indicates the number

of factors that should be generated by the analysis. The first three eigenvalues form a steep curve, followed by a bend and then a straight-line trend with shallow slope[18]. The recommendation is to retain those eigenvalues in the steep curve before the first one on the straight line[18]. The scree plot confirmed an earlier conclusion made by the eigenvalues that 3 principal components was enough to explain variations from original data which was about 93% in total, Figure 18.

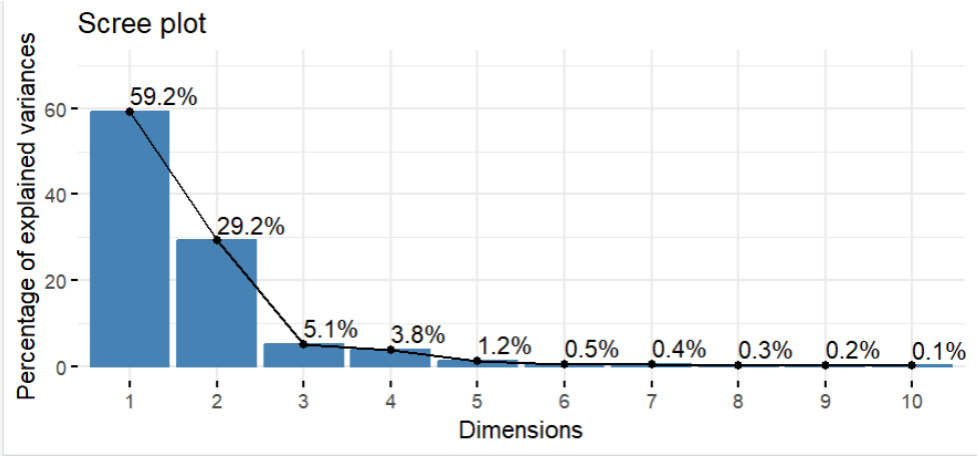


Figure 18: Scree plot.

There was need to clearly specify which variables contributed most to the three principal components. From the eigenvector's matrix, the first principal component Prin1, was written as a linear combination of the original variables,

$$Prin1 = 0.059936V_1 - 0.00529V_3 + ... + 0.0026811V_{110}$$

The second principal component Prin2 was,

$$Prin2 = 0.013899V_1 + 0.028365V_3 - ... + 0.127930776V_{110}$$

The third principal component Prin3 was,

$$Prin3 = 0.098596V_1 + 0.015988V_3 + ... - 0.110117110V_{110},$$

where the variables were standardized, Figure 19.

	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9	Comp.10	Comp.11	Comp.12	Comp.13
V1	0.059936	0.013899	0.098596	0.039706	0.088272	0.024282	0.08729	0.031543	0.018988	0.182261	0.201359	0.030312	0.884405
V3	-0.00529	0.028365	0.015988	-0.03826	-0.16493	-0.44611	0.031814	-0.02937	0.050275	0.098523	0.044714	0.077517	-0.22765
V5	0.007628	-0.01319	0.026307	-0.03373	-0.04766	-0.51396	-0.02542	0.118206	0.01911	-0.06473	-0.05446	-0.04229	0.122491
V6	0.007628	-0.01319	0.026307	-0.03373	-0.04766	-0.51396	-0.02542	0.118206	0.01911	-0.06473	-0.05446	-0.04229	0.122491
V7	0.049973	-0.13923	-0.02041	0.108949	-0.0664	-0.16517	0.039386	0.072544	0.001193	0.062608	-0.03783	0.073591	0.052799
V8	0.049935	-0.12821	-0.00069	0.085804	-0.04486	-0.24261	0.064722	0.069361	-0.0232	-0.0382	-0.07535	-0.04072	0.038845
V9	0.047039	-0.13852	-0.00087	0.127478	-0.06347	-0.17779	-0.02236	0.107918	-0.02046	-0.01551	-0.04317	-0.03292	-0.01724
V10	0.02553	-0.14133	-0.06521	0.093348	-0.13137	-0.0019	0.147925	-0.01783	0.032369	0.140441	0.095314	0.114462	-0.04334
V11	0.026454	-0.13697	-0.05263	0.076625	-0.13887	-0.16944	0.13057	0.021819	0.036619	0.110723	0.071711	0.093655	-0.00073
V12	0.052819	-0.14001	0.008522	0.167903	-0.04349	0.067743	0.08036	0.033829	-0.10119	-0.04231	-0.06001	-0.03406	-0.00035
V13	0.049704	-0.1544	0.000536	0.139588	-0.04481	-0.02491	0.024428	0.019028	-0.04128	-0.01357	-0.04935	-0.07387	-0.03738
V14	-0.04394	0.163014	0.052093	-0.08698	0.003082	0.01792	0.060136	0.050473	-0.16393	-0.03604	-0.01765	0.051928	0.054845
V15	0.052606	-0.14021	0.006926	0.16802	-0.04609	0.063315	0.078841	0.037845	-0.10089	-0.03979	-0.06149	-0.03203	-0.00422
V16	-0.02338	0.141831	0.032911	-0.04553	-0.06897	-0.06162	0.130058	0.116002	-0.26806	0.040945	-0.1006	0.10564	-0.04947
V17	0.05286	-0.14035	0.008234	0.16766	-0.04442	0.063743	0.079691	0.034933	-0.09954	-0.04255	-0.0601	-0.03363	-0.00142
V18	-0.06007	-0.12509	0.154368	-0.08362	0.124038	-0.01808	0.070553	0.020893	-0.00066	0.049693	-0.04694	-0.02525	-0.06206
V19	0.050278	-0.06557	0.06569	-0.03558	0.289574	0.000286	0.132502	0.23404	0.135068	0.02502	-0.14652	0.117574	-0.02934
V20	0.064786	-0.07225	-0.08413	0.194277	-0.10988	0.048463	-0.05938	0.09745	-0.08361	-0.0238	0.167471	0.189512	0.00503
V21	0.153887	0.053478	-0.02656	0.02709	-0.01572	-0.00462	-0.02518	-0.05162	0.02294	-0.09962	-0.05019	0.092303	0.023025
V22	0.077803	0.129197	-0.06978	0.072544	0.033134	-0.0082	0.15203	0.001982	0.076673	0.00391	-0.21472	-0.22126	0.022822
V23	-0.02615	-0.12589	-0.05613	-0.1629	-0.01135	-0.02788	0.077046	-0.08214	-0.08102	0.207643	0.248051	-0.28928	-0.065
V24	0.076082	-0.04763	0.040722	0.016144	0.092086	-0.01313	0.098688	-0.19651	-0.04249	0.488551	-0.18096	0.369577	-0.05048

Figure 19: Feature loadings for 13 principal components.

Since the main objective of the research was to end up with less number of variables, based on three principal components chosen, a low loadings filter was used based on the five number summary for each variables in the three principal components, a threshold value of 0.1 was chosen and features with loadings below 0.1 removed. This finally resulted to 39 features out of the 110 in the original data set. A summary of the selected features were as follows; principal component one had a total of 17 features whereby 1 was intensity and 16 were texture based features, principal component two had a total of 13 features whereby 2 were shape, 4 intensity and 7 texture based features, finally, principal component three had a total of 9 features whereby 1 was shape, 3 intensity and 5 texture based features. From our findings, We concluded the most important feature category based on PCA was texture feature category. A summary of selected PCA features, Table 3.

Principal Component 1	Principal Component 2	Principal Component 3
original_firstorder_Minimum_CT– intensity feature	original_shape_MinorAxisLength_CT – shape feature	original_shape_VoxelVolume_CT– shape feature
original_glcm_Correlation_CT– texture feature	original_shape_SurfaceArea_CT – shape feature	original_firstorder_MeanAbsoluteDeviation_CT– intensity featuresn
original_glcm_DifferenceAverage_CT – texture feature	original_firstorder_Energy_CT – intensity feature	original_firstorder_Median_CT– intensity feature
original_glcm_JointEnergy_CT– texture feature	original_firstorder_Maximum_CT – intensity feature	original_firstorder_RootMeanSquared_CT– intensity feature
original_glcm_MCC_CT– texture feature	original_firstorder_Range_CT – intensity feature	original_glcm_ClusterProminence_CT– texture feature
original_gldm_DependenceNonUniformityNormalized_CT – texture	original_firstorder_RobustMeanAbsoluteDeviation_CT – intensity feature	original_glrlm_RunLengthNonUniformity_CT– texture feature
original_gldm_LowGrayLevelEmphasis_CT– texture feature	original_glcm_ClusterTendency_CT – texture feature	original_glrlm_RunLengthNonUniformityNormalized_CT– texture feature
original_gldm_SmallDependenceEmphasis_CT – texture feature	original_glcm_DifferenceEntropy_CT – texture feature	original_glrlm_RunVariance_CT– texture feature
original_glrlm_LowGrayLevelRunEmphasis_CT – texture feature	original_glcm_Imc1_CT – texture feature	original_glszm_ZoneEntropy_CT– texture feature
. original_glrlm_RunLengthNonUniformity_CT– texture feature	. original_glcm_JointEntropy_CT – texture feature	
original_glrlm_RunLengthNonUniformityNormalized_CT – texture feature	original_ngtdm_Busyness_CT – texture feature	
original_glrlm_RunVariance_CT – texture feature	original_ngtdm_Complexity_CT – texture feature	
original_glszm_SizeZoneNonUniformity_CT– texture feature	original_ngtdm_Contrast_CT– texture feature	
original_glszm_SizeZoneNonUniformityNormalized_CT– texture feature		
original_glszm_SmallAreaEmphasis_CT– texture feature		
original_glszm_ZoneEntropy_CT– texture feature		
original_ngtdm_Coarseness_CT– texture feature		

Figure 20: A list of features selected in each principal component.

Reduction Technique	Principal Component	Categories	Number of variables selected	Percentage
Principal Component Analysis (PCA)	PC1	Intensity	1	2.6
		Shape	0	0
		Texture	16	41
	PC2	Intensity	2	5.2
		Shape	4	10.3
		Texture	7	17.9
	PC3	Intensity	1	2.6
		Shape	3	7.7
		Texture	5	12.8
		Total	39	100

Table 4: Features selected through PCA.

A heat map is a visual representation of data in which values are represented as colors. These maps use a color gradient to represent the values with cooler colors e.g green or blue indicating low values and warm colors e.g orange or red indicating high values. R program was used to generate the clustered heat map since it's goal is to build associations between features selected. A report on the correlation between variables indicated very strong correlation in the main diagonal. It was also observed that there was a very weak correlation between most features on the west of the map and features to the north eastern part of the heat map.

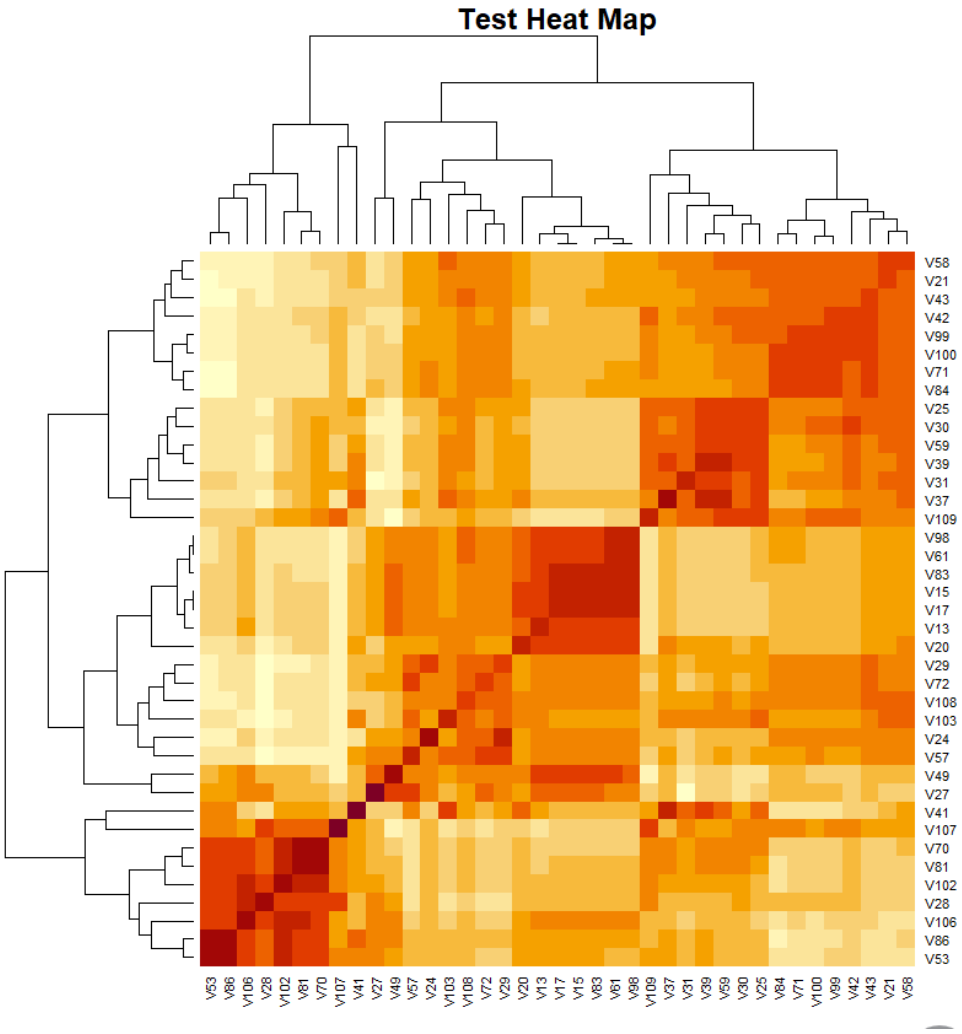


Figure 21: Heat map of selected features based on PCA.

4.2 Clustering Analysis

4.2.1 Hierarchical Clustering

Examining the agglomerative hierarchical approach on the extracted features from lung cancer data by complete, average and ward's minimum-variance clustering methods, R program was used. A dendrogram is a graphical representation of the hierarchy of clusters shows the distance between clusters and the order in which they were merged. The height of the branches on the dendrogram represents the distance between the clusters. The closer the branches are to each other, the more similar the clusters are, Figures 22, 23 and 24.

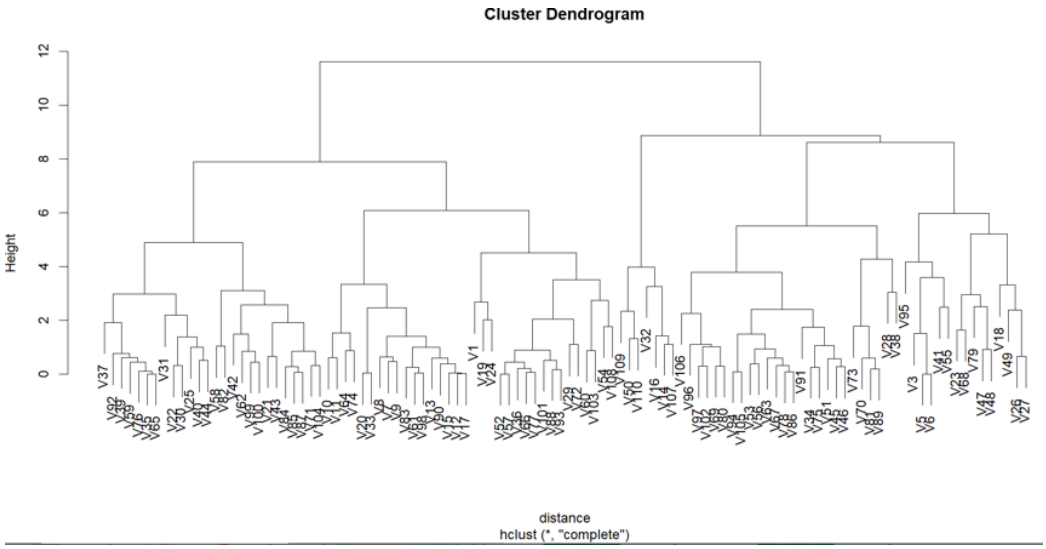


Figure 22: Cluster dendogram of all features.

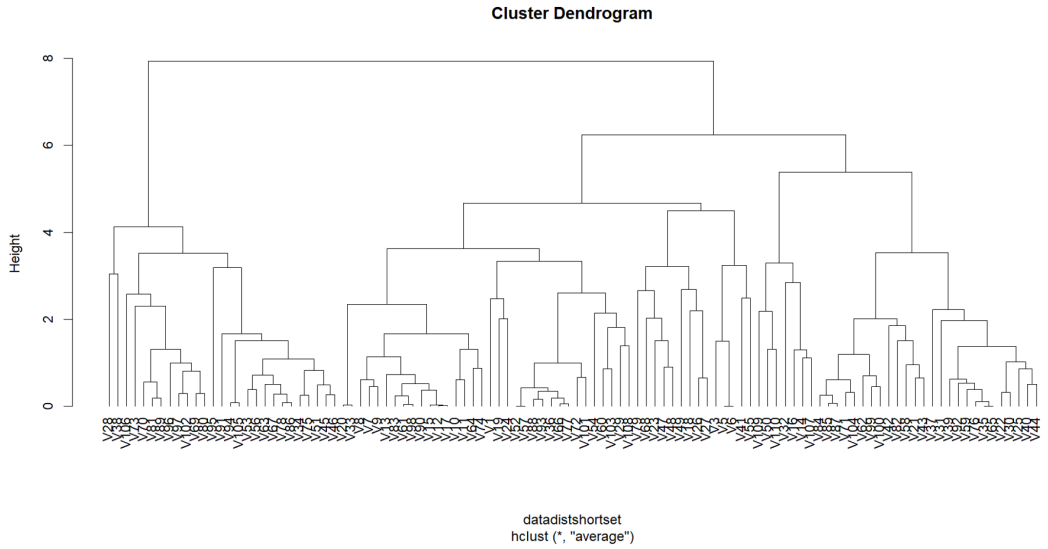


Figure 23: Cluster dendogram of all features.

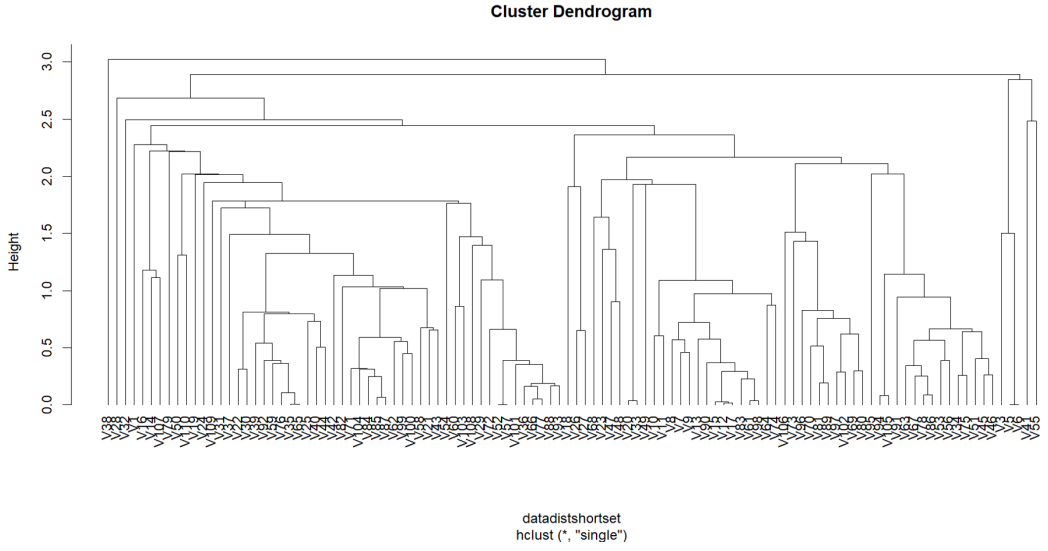


Figure 24: Cluster dendogram of all features.

The hierarchical clustering algorithm grouped the features into 3 clusters, Table 5.

Features	Cluster 1	Cluster 2	Cluster 3
V1	1	0	0
V10	0	1	0
V100	1	0	0
V101	0	1	0
V102	0	0	1
V103	0	1	0
V104	1	0	0
V105	0	0	1
V106	0	0	1
V107	0	0	1
V108	0	1	0
V109	1	0	0
V11	0	1	0
V110	1	0	0
V12	0	1	0
V13	0	1	0
V14	0	0	1
V15	0	1	0
V16	0	0	1
V17	0	1	0
V19	0	1	0
V20	0	1	0
V21	1	0	0
V23	0	1	0
V24	0	1	0
V25	1	0	0
V26	0	1	0
V27	0	1	0
V28	0	0	1
V29	0	1	0

Table 5: Features per cluster.

The feature names were $V_1, \ldots, V_{110}$ as opposed to the initial names of the vari-

ables, It was observed that cluster 2 started grouping with feature V_{10} then V_{101} as it was trying to cluster features with the closest distance and the iteration continued till all the features were clustered. After determining the number of clusters, the three clusters were further analyzed to understand their characteristics. This was realized by identifying variables in each cluster and identifying any patterns or similarities. A 'cutree()' function on R grouped 48 features in cluster 1, 33 features in cluster 2 and 27 features in cluster 3.

Optimal number of clusters

To diagnose if the number of clusters chosen were adequate, R was used to generate;

a) A silhouette plot which is a graph used to interpret the results of clustering algorithms, including hierarchical clustering. In this plot, each data point is represented by a vertical line or bar that reflects its silhouette coefficient. This is a measure of how similar the point is to its own cluster compared to other clusters. The silhouette coefficient ranges from -1 to 1, with higher values indicating that the point is well-matched to its own cluster and poorly matched to other clusters. The silhouette plot typically shows each data point sorted by its cluster assignment, with the silhouette coefficient plotted against the index of the data point, Figures 25, 26 and 27.

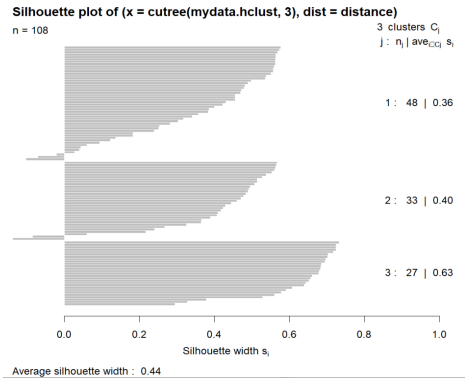


Figure 25: Silhouette plot 3 clusters.

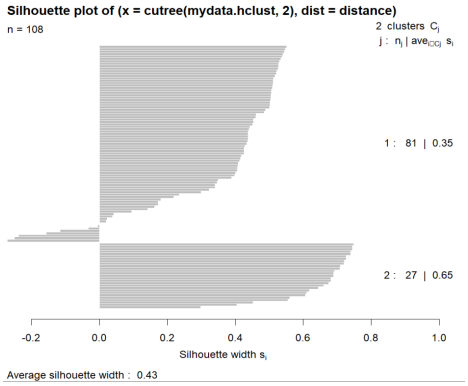


Figure 26: Silhouette plot 2 clusters.

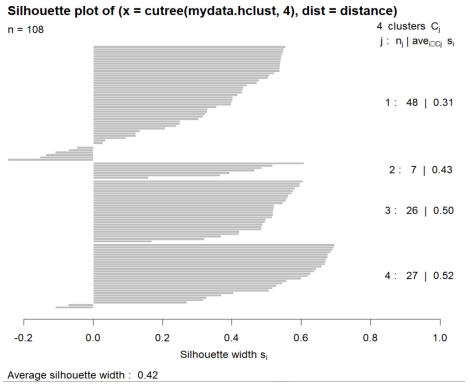


Figure 27: Silhouette plot 4 clusters.

Interpretation of silhouette plots considers the following;

- Looking for well-defined clusters: If there are clear gaps between the bars for different clusters, this indicates that the clustering algorithm has successfully separated the data into distinct groups. For our silhouette plots, Figure 25 and 26 have clear gaps while Figure 27 has no clear gap showing the clustering algorithm did not successfully separate the clusters.
- Looking for an overlap between clusters: If there is significant overlap between the bars for different clusters, this indicates that the clustering algorithm may not have successfully separated the data into distinct groups. Figure 27 had an overlap, while Figure 25 and 26 had no overlaps making the two better clusters.
- Finally, assess the overall quality of the clustering: The average silhouette coefficient for all data points can be used as a measure of the overall quality of the clustering. A high average silhouette coefficient indicates that the data points are well-matched to their respective clusters, while a low average silhouette coefficient indicates that the clustering algorithm may not have successfully separated the data into distinct groups. So out of the clustering, the plot with 3 clusters has a silhouette coefficient of 4.3 making it a more appropriate number of clusters.

b) Elbow Method looks at the total within-cluster sum of squares (wss) as a fucntion of the number of clusters. Just like is in the case of a scree plot in the PCA, the location of a knee is considered an indicator of appropriate number

of clusters. Based on the plot it was noted that 3 clusters would suffice, Figure 28.

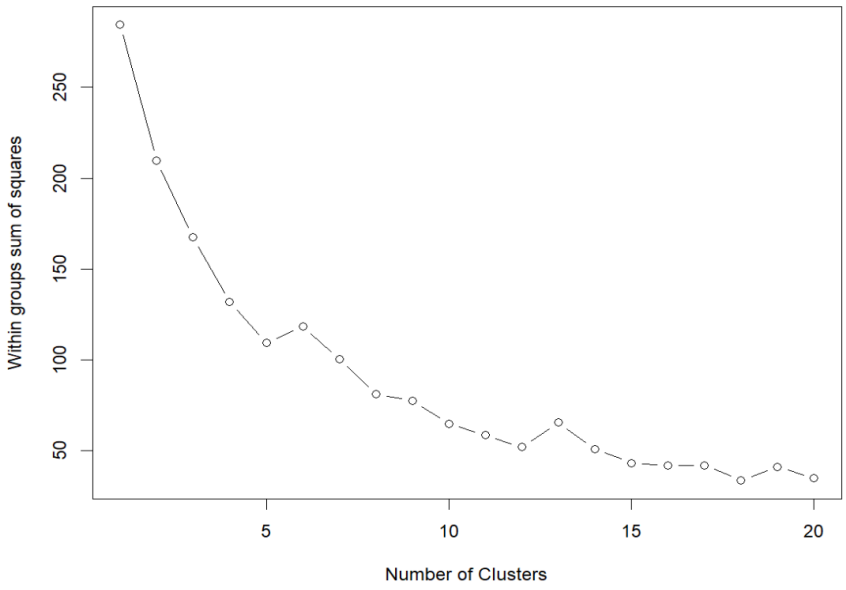


Figure 28: Elbow method for optimal number of clusters.

The analysis progressed into using a variance filter on each cluster to identify features within each cluster with the most similarity. A five number summary on the selected feature matrix identified the 25^{th} percentile averaging as 0.01 as an adequate threshold which was to drop every feature with a variance higher than 0.02. A summary of the selected features were as follows 21 features; 4 in Clst1, 3 in Clst2 and 14 in Clst3. Cluster one features consisted of 1 intensity, 2 shape and 1 texture based features, Cluster two had a total of 3 features whereby 2 were intensity, 1 shape and no texture based features, cluster three however had 14 texture based features with no shape nor intensity based features. The findings concluded that texture based

features are the most dominant selected features contributing the largest percentage to the new data matrix. A summary of the findings, Table 5.

Reduction Technique	Clusters	Categories	Number of Variables selected	Percentage
Hierarchical Clustering	Clst1	Intensity	1	4.7
		Shape	2	9.5
		Texture	1	4.7
	Clst2	Intensity	2	9.5
		Shape	1	4.7
		Texture	0	0
	Clst3	Intensity	0	0
		Shape	0	0
		Texture	14	67
		Total	21	100

Table 6: Summary table of features selected through hierarchical clustering.

A list of specific features in each cluster were identified and listed, Table 6.

4.2.2 k-means

For the k-means analysis it is very important to find the optimal number of clusters before hand before doing our analysis on the data set. R software was used in the analysis. The analysis was preceded by loading two libraries; library(factoextra) and library(cluster). The cancer data that had an earlier been loaded was assigned the name df, R through the function na.omit(df) deleted rows with missing values which is a data science management technique. Each variable was scaled to have a mean of 0 and standard deviation of 1 using the function 'scale(df)'. To find the optimal number of clusters two plots were used:

Clusters	Cluster 1	Cluster 2	Cluster 3
Selected Features	diagnosticImage.original MaximumCT-intensity feature	originalshapeSurface VolumeRatioCT-shape feature	originalglcmCluster ShadeCT-texture feature
	originalshapeFlatnessCT -shape feature	originalfirstorderInterquartile RangeCT-intensity feature	originalgldmLargeDependence LowGrayLevelEmphasisCT-texture feature
	originalshapeLeastAxisLength CT-shape feature	originalfirstorderRobustMean AbsoluteDeviationCT-intensity feature	originalgldmLowGrayLevel EmphasisCT-texture feature
	originalgldmLargeDependence HighGrayLevelEmphasisCT texture feature		originalglrmLongRunGray LevelEmphasisCT-texture feature
			originalglrmLowGrayLevel RunEmphasisCT-texture feature
			originalglrmShortRunLow GrayLevelEmphasisCT-texture feature
			originalglszmGrayLevelNon UniformityNormalizedCT-texture feature
			originalglszmLargeArea EmphasisCT-texture feature
			originalglszmLargeAreaHigh GrayLevelEmphasisCT-texture feature
			originalglszmLargeAreaLow GrayLevelEmphasisCT-texture feature
			originalglszmLowGrayLevel ZoneEmphasisCT-texture feature
			originalglszmSmallAreaLow GrayLevelEmphasisCT-texture feature
			originalglszmZoneVarianceCT texture feature

Table 7: List of features selected under hierarchical clustering.

a) Number of Clusters vs. the Total Within Sum of Squares which involves the use of the $fviz\_nbclust()$ function to create a plot of the number of clusters versus the total within sum of squares. Our resulting plot from this method is shown in figure 25. When we create this type of plot we look for an "elbow" where the sum of squares begins to "bend" or level off. This is typically the optimal number of clusters. For this plot it appears that there is a bit of an elbow or "bend" at k = 3, Figure 29.

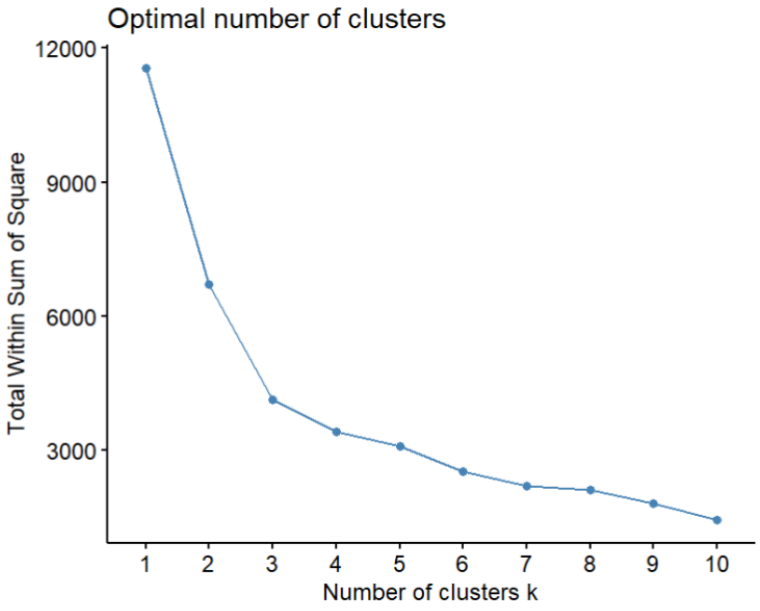


Figure 29: A plot of the total within sum of squares vs. the number of clusters.

b) The other method to identify the optimal number of clusters is to use a metric known as the gap statistic. This method compares total intra-cluster variation for different values of k with their expected values for a distribution with no clustering. Gap statistic was calculated for each number of clusters using the $clusGap()$ function from the cluster package along with a plot of clusters vs. gap statistic using the $fviz\_gap\_stat()$ function: From the plot it was observed that the gap statistic was highest at k = 3 clusters, Figure 30.

Clustering analysis on the data set by categorizing the objects into 3 clusters respectively made it possible to visualize the respective features in each cluster through a cluster plot and cluster mapping generated through R, Figure 31 and 32.

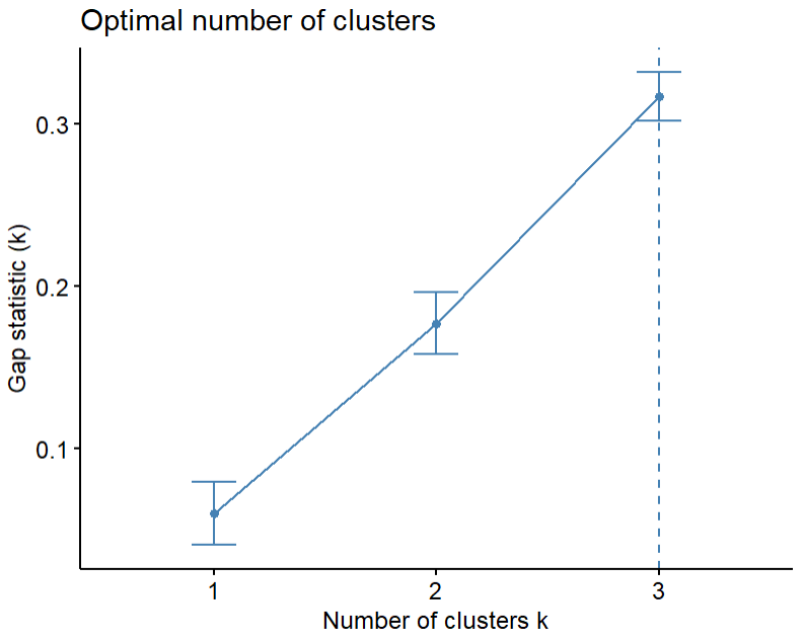


Figure 30: A plot of gap statistic versus clusters.

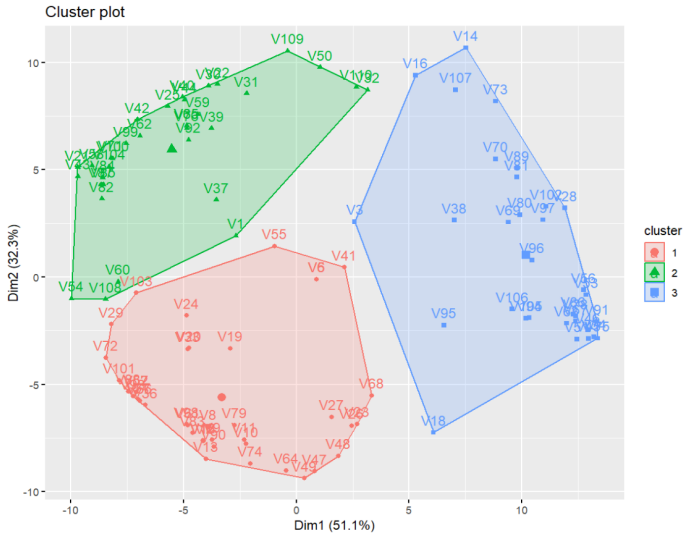


Figure 31: A cluster plot of the 3 optimum clusters of features under k-means clustering.

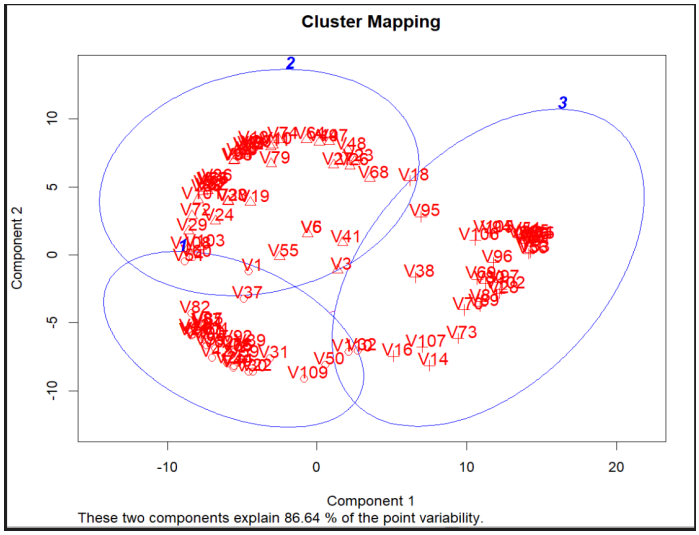


Figure 32: A cluster mapping of of the 3 optimum clusters of features under k-means clustering.

A variance selection filter on each cluster on k-means clustering was then done.A variance threshold of 0.02, was used removed every feature with a variance less than 0.02 and finally ended up with 21 variables; 3 in Clst1, 3 in Clst2 and 15 in Clst3.

Cluster one features consisted of 3 intensity based features and no shape nor texture based features, Cluster two had a total of 3 features whereby 1 intensity,1 shape and 1 texture based features, cluster three no intensity, 1 shape and 14 texture based features. Findings concluded the most important feature category based on k-means clustering analysis were the texture based features, Table 7.

The specific features per cluster, Table 8

Reduction Technique	Clusters	Categories	Number of Variables Selected	Percentage
k-meansclustering	Clst1	Intensity	3	14
		Shape	0	0
		Texture	0	0
	Clst2	Intensity	1	4.7
		Shape	1	4.7
		Texture	1	4.7
	Clst3	Intensity	0	0
		Shape	1	4.7
		Texture	14	67
		Total	21	100

Table 8: Summary table of features selected through k-means clustering.

Clusters	Cluster 1	Cluster 2	Cluster 3
Selected Features	diagnosticImage.original MaximumCT-intensity feature	origin alshapeFlatness shape feature	originalShapeSurfaceVolume RatioCT-shape feature
	originalfirstorderInterquartile RangeCT-intensity feature	originalshapeLeastAxis LengthCT-intensity feature	originalglcmClusterShadeCT -texture feature
	originalfirstorderRobustMean AbsoluteDeviationCT- intensity feature	originalgldmLargeDependence HighGrayLevelEmphasisCT- texture feature	originalgldmLargeDependence LowGrayLevelEmphasisCT- texture feature
			originalgldmLowGrayLevel EmphasisCT-texture feature
			originalglrmLongRunLow GrayLevelEmphasisCT-texture feature
			originalglrmShortRunLowGray LevelEmphasisCT-texture feature
			originalglszmGrayLevelNon UniformityNormalizedCT -texture feature
			originalglszmLargeArea Emphasis-CTtexture feature
			originalglszmLargeAreaHigh GrayLevelEmphasisCT-texture feature
			originalglszmLowGrayLevel ZoneEmphasisCT-texture feature
			originalglszmSmallAreaLow GrayLevelEmphasisCT- texture feature
			originalglszmZoneVarianceCT texture feature
			originalngdtmZoneBusynessCT texture feature

Table 9: List of features selected under k-means clustering.

4.3 Isometric Feature Mapping (ISOMAP)

The research intention for using this non-linear dimensionality reduction technique was to map high-dimensional data onto a lower-dimensional space while preserving the underlying structure of the data. To realize this R program was used. The procedure involved installing and running a library called 'vegan' in R. The number of axes in metric scaling using the argument 'k' in cmdscale(Classical (Metric) Multidimensional Scaling scale) was specified, the analysis then inputed the number of shortest dissimilarities retained for a point to be 'k=3' to use the three nearest neighbors to construct the geodesic distance matrix, which was later opted to use 'epsilon=0.45', finally the 'isomap()' function which under 'MASS' package in R was used.To visualize the results 'plot()' function was used. The resulting plot showed the algorithm was able to separate the different features into distinct clusters in the lower dimensional space, Figure 34 and 35 ISOMAP was able to preserve the underlying structure of the data and capture the important features that differentiate the different features in each of the clusters. It is however, unfortunate the results could not clearly tell which specific features for each of the clusters were chosen in the lower dimensional space.

More expertise and skill would be necessary to successfully analyze using this kind of method and give a clear report of the same. Having achieved this objective using PCA, clustering Analysis augmented with either loadings or variance filter, it was not very urgent to apply detailed analysis, however this was recommended as a great starting point for future work. Results for this analysis was visualized in a 2-D space, Figure 33 and 34.

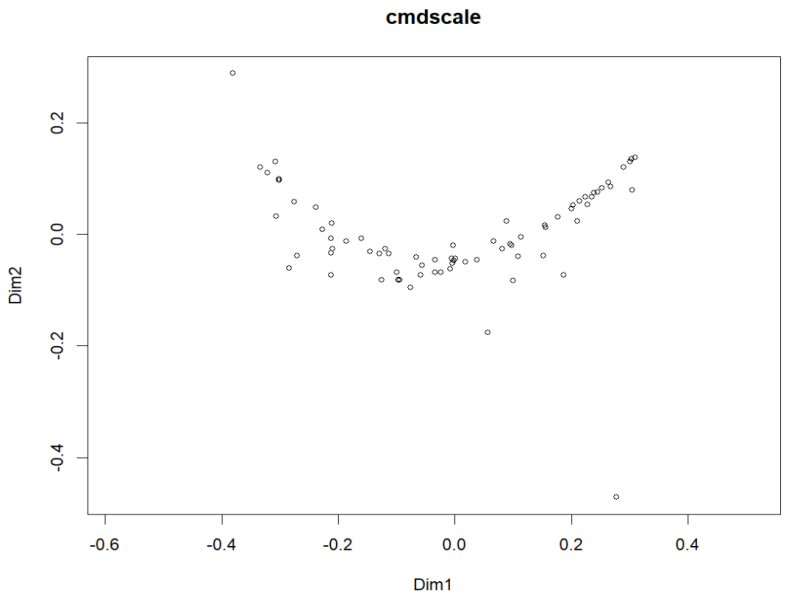


Figure 33: A 2-D plot to visualizing how similar each features are across all of the variables in the data frame.

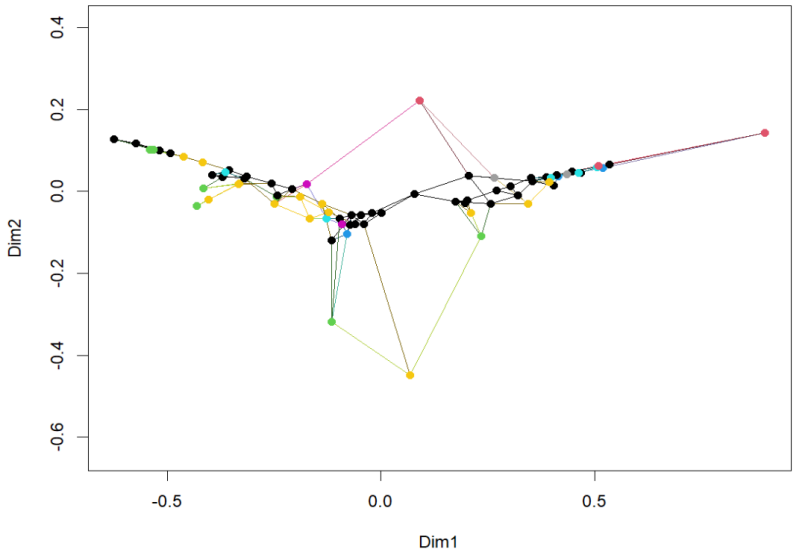


Figure 34: A 2-D coloured plot to visualizing how similar each features are across all of the variables in the data frame.

5 CONCLUSION

The conclusion of any research is to ascertain if the research questions asked at the beginning of the research were successfully answered. This particular research was to find out if the number of variables could be reduced from 110 to a lesser number. After analysis, 39 features were selected for PCA, 21 features for both Hierarchical and k-means clustering. The other research question we intended to answer was to ascertain the most significant feature category, all the analyses confirmed that the texture based feature category was the most important features. The intended goal of the research objective was met as we were able to reduce the features to a number averaging between 20 to 40 which approximately reduced our dimension by over 75% approximately to a data matrix of 74 by 39 for through PCA and 74 by 21 for clustering analyses.

To be specific, for PCA the eigenvalues indicated that three principal components provided a good summary of the data accounting for 93% of the total variance while the rest of the components only accounted for less 4% each. 39 features were then selected based on the loadings filter with a threshold value of 0.1 that had been computed and agreed on based on the five- number summary. Principal component one had a total of 17 features whereby 1 was intensity and 16 were texture based features, principal component two had a total of 13 features whereby 2 were shape, 4 intensity and 7 texture based features, finally, principal component three had a total of 9 features whereby 1 was shape, 3 intensity and 5 texture based features. For clustering analysis, agglomerative hierarchical clustering algorithm clustered the features to 3 clusters. Selection of 21 features based on the variance filter with

a 0.02 threshold value was done , whereby 3 were intensity, 3 shape and 15 were texture based features. k-means clustering algorithm with an initial cluster optimum cluster of 3, selected 21 features based on the variance filter with a threshold value we computed of 0.02, out of which 4 intensity, 1 shape and 15 texture based features were selected. ISOMAP analysis generated a graph with variables clustered in different classes indicated by black, yellow, green, purple, blue, red, grey and turquoise colors, however it was hard to clearly tell which specific variable was in which cluster and how many were selected. Overally, all our analyses clearly outlined the texture based features as the most significant features in the lung cancer data. The texture feature as earlier discussed is in the Second order radiomics category, it is concerned with texture features and relations between pixels to model intra-tumor heterogeneity. These features are generated from different matrices such as GLCM, GLRLM, NGTDM and finally GLZLM[26].

6 FUTURE WORK

There is still more to the future of this data exploration. By shear fact that the end of research is able to reduce the data into a manageable matrix, our objective should not stop at this however, future works may therefore explore prognosis and therapy on the cancer patients. We can do this by using the significant variables chosen which are 21 for clustering or 35 for PCA as new predictors to;

i) Perform logistic regression analysis to compare cancer stages among males and females.

ii) Perform multinomial logistic regression analysis to predict the cancer stage of a patient.

An in-depth analysis into using Manifold logarithms such as ISOMAP and Laplacian to counter check the adequacy in the number of features selected and ascertain if the texture based features are the most important is equally very necessary. Being that cancer is a pressing global health concern being a leading cause of death worldwide, timeliness of detection and diagnosis is critical to maximizing the chances of successful treatment. The better we understand these diseases, the more progress we will make toward diminishing the tremendous human and economic tolls of cancer.

BIBLIOGRAPHY

[1] Velazquez ER Aerts HJ, Leijenaar RT, Parmar C, Grossmann P, Carvalho S, Bussink J, Monshouwer R, Haibe-Kains B, Rietveld D, Hoebers F, Rietbergen MM, Leemans C R, Dekker A, Quackenbush J, Gillies RJ, and Lambin P, *Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach*, Nat Commun (2014), 5–4644.

[2] Parnian Afshar, Arash Mohammadi, Konstantinos N Plataniotis, Anastasia Oikonomou, and Habib Benali, *From handcrafted to deep-learning-based cancer radiomics: challenges and opportunities*, IEEE Signal Processing Magazine **36** (2019), no. 4, 132–160.

[3] Mikhail Belkin and Partha Niyogi, *Laplacian eigenmaps for dimensionality reduction and data representation*, Neural Computation **15** (2003), no. 6, 1373–1396.

[4] Michael Berry and Azlinah Mohamed, *Supervised and unsupervised learning for data science*, 01 2020.

[5] B. Chen, L. Yang, R. Zhang, W. Luo, and W. Li, *Radiomics: an overview in lung cancer management—a narrative review*, Annals of Translational Medicine **8** (2020), 1191.

[6] Kapsoulis D., Tsiakas K., Trompoukis X., Asouti V., and Giannakoglou K., *A pca-assisted hybrid algorithm combining eas and adjoint methods for cfd-based optimization*, Applied Soft Computing **73** (2018), 520–529.

[7] Andriy Fedorov, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-Christophe Fillion-Robin, Sonia Pujol, Christian Bauer, Dominique Jennings, Fiona Fennessy, Milan Sonka, John Buatti, Stephen Aylward, James V. Miller, Steve Pieper, and Ron Kikinis, *3d slicer as an image computing platform for the quantitative imaging network*, Magnetic Resonance Imaging **30** (2012), 1323–1341.

[8] Wu G., Jochems A., Refaee T., Ibrahim A., Yan C., Sanduleanu S., Woodruff H. C., and Lambin P, *Structural and functional radiomics for lung cancer*, European Journal of Nuclear Medicine and Molecular Imaging **48** (2021), 3961–3974.

[9] Joost JM Van Griethuysen, Andriy Fedorov, Chintan Parmar, Ahmed Hosny, Nicole Aucoin, Vivek Narayan, Regina GH Beets-Tan, Jean-Christophe Fillion-Robin, Steve Pieper, and Hugo JWL Aerts., *Computational radiomics system to decode the radiographic phenotype*, Cancer research **77** (2017), e104–e107.

[10] Liang H., Sun X., Sun Y., and Gao Y., *Text feature extraction based on deep learning: a review.*, EURASIP Journal on Wireless Communications and Networking (2017), 211.

[11] Pascal Hannequin, Chantal Decroisette, Pascale Kermanach, Giulia Berardi, and Vincent Bourbonne, *Fdg pet and ct radiomics in diagnosis and prognosis of non-small-cell lung cancer*, Transl Lung Cancer Res **11** (2022), 2051–63.

[12] Yu-Ming Huang, Tsang-En Wang, Ming-Jen Chen, Ching-Chung Lin, Ching-Wei Chang, Hung-Chi Tai, Shih-Ming Hsu, and Yu-Jen Chen, *Radiomics-based*

nomogram as predictive model for prognosis of hepatocellular carcinoma with portal vein tumor thrombosis receiving radiotherapy, Frontiers in Oncology **12** (2022).

[13] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, *An introduction to statistical learning*, vol. 112, Springer, 2013.

[14] Virendra Kumar, Yuhua Gu, Satrajit Basu, Anders Berglund, Steven A. Eschrich, Matthew B. Schabath, Kenneth Forster, Hugo J.W.L. Aertsf, Andre Dekker, David Fenstermacher, Dmitry B. Goldgof, Lawrence O. Hall, Philippe Lambin, Yoganand Balagurunathan, Robert A. Gatenby, and Robert J. Gillies, *Radiomics: the process and the challenges*, Magn Reson Imag **30** (2012), 1234–48.

[15] Li Loraine., *Principal component analysis for dimensionality reduction.*, Medium; Towards Data Science (2019).

[16] Zhou M., Scott J.and Chaudhury B.and Hall L., Goldgof D.and Yeom K. W.and Iv M.and Ou Y.and Kalpathy-Cramer, J. Napel, S. Gillies, R. Gevaert, O., and Gatenby R., *Radiomics in brain tumor: Image assessment, quantitative feature descriptors, and machine-learning approaches and machine-learning approaches*, American Journal of Neuroradiology **39** (2017), 208–216.

[17] J MacQueen, *Classification and analysis of multivariate observations*, 5th Berkeley Symp. Math. Statist. Probability (1967), 281–297.

[18] Zahed Mostafa and Skafyan Maryam, *Application of feature selection and dimension reduction techniques on large-scale ct dataset for lung cancer diagnosis based on radiomics-sesug 2022 paper 222*, (2022), 5–13.

[19] Lambin P, Rios-Velazquez E, Leijenaar R, Carvalho S, van Stiphout RG, Granton P, Zegers CM, Gillies R, Boellard R, Dekker A, and Aerts HJ, *Radiomics: extracting more information from medical images using advanced feature analysis*, Eur J Cancer **48** (2012), 441–6.

[20] Young-Jin Park and Hui-Sup Cho, *Lung cancer tumor detection method using improved ct images on a one-stage detector*, (2022).

[21] P. Ray, S. S. Reddy, and T. Banerjee, *Various dimension reduction techniques for high dimensional data analysis: a review*, Artificial Intelligence Review **54** (2021), 3473–3515.

[22] C Rencher Alvin, *Methods of multivariate analysis, john wiley&sons inc*, Publication, Canada (2002).

[23] Gillies RJ, Kinahan PE, and Hricak H, *Radiomics: Images are more than pictures, they are data*, Radiology **278** (2016), 563–77.

[24] Kai Sun, Jing Zhang, Zhenyu Liu, Qi Qiu, Han Gao, Panpan Liu, Kuntao Chen, Wei Wei, Liang Wang, Junting Zhang, et al., *A deep learning radiomics analysis for identifying sinus invasion in patients with meningioma before operation using tumor and peritumoral regions*, European Journal of Radiology **149** (2022), 110187.

[25] Joshua B Tenenbaum, Vin de Silva, and John C Langford, *A global geometric framework for nonlinear dimensionality reduction*, science **290** (2000), no. 5500, 2319–2323.

[26] Jie Tian, Di Dong, Zhenyu Liu, and Jingwei Wei, *Radiomics and Its clinical application*, The Elsevier and Miccai Society Book Series, Mara Conner, July 2021.

[27] Jia Weikuan, Sun Meili, Lian Jian, and Hou Sujuan, *Feature dimensionality reduction: a review*, Complex & Intelligent Systems **8** (2022), 2198–6053.

[28] Liu Z., Wang S., Dong D., Wei J., Fang C., Zhou X., Sun K., Li L., Li B., Wang M., and Tian J., *The applications of radiomics in precision diagnosis and treatment of oncology: Opportunities and challenges.*, Theranostics **9** (2019), 1303–1322.

[29] Wang Z., Yang C., Han W., Sui X., Zheng F., Xue F., Xu X., Wu P., Chen Y., Gu W., W. Song, and Jiang J., *Quantifying lung cancer heterogeneity using novel ct features a cross-institute study*, Insights into Imaging **13** (2022).

APPENDICES

1 CODES

1.1 R codes

```
######################
install.packages("readxl")
library(readxl)
file.choose()
Lung_CT<- read_excel("C:\\Users\\Student175
\\Desktop\\spring 2023\\Thesis\\1. Dissertation
\\Data sets\\Raw data\\Lung_Cancer_CT.xlsx")
View(Lung_CT)

#define Min-Max normalization function
min_max_norm <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}

#apply Min-Max normalization to dataset
LungCT_norm <- as.data.frame(lapply(Lung_CT,
min_max_norm))
```

```
write.csv(LungCT_norm,"C:/Users/Student175/
Desktop/spring 2023/Thesis/1. Dissertation/
Data sets/Raw data/LungCT_norm.csv")

file.choose()
LungCT_norm<- read.csv("C:\\Users\\Student175
\\Desktop\\spring 2023\\Thesis\\1. Dissertation\\
Data sets\\Raw data\\LungCT_norm.csv")
LungCT_norm<-as.matrix(LungCT_norm)
View(LungCT_norm)

##data prep
Lung_Norm_mod<-LungCT_norm[ , which(apply
(LungCT_norm, 2, var) != 0)]
View(Lung_Norm_mod)
write.csv(Lung_Norm_mod,"C:/Users/Student175
/Desktop/spring 2023/Thesis/1. Dissertation/Data
sets/Raw data/Lung_Norm_mod.csv")

##Correlation matrix
Lung_Norm_mod<-cor(Lung_Norm_mod)
View(Lung_Norm_mod)
dim(Lung_Norm_mod)
```

```
##1. PCA##
install.packages("factoextra")
library("factoextra")
library("FactoMineR")

res.pca <- princomp(Lung_Norm_mod)
fviz_eig(res.pca)

### Extract and visualize eigenvalues/variances:

# Extract eigenvalues/variances
get_eig(res.pca)
Eigenvalues <- as.data.frame(get_eig(res.pca))
write.csv(Eigenvalues,"C:/Users/Student175/
Desktop/spring 2023/Thesis/1. Dissertation/
Data sets/Raw data/Eigenvalues.csv")
View(Eigenvalues)

# Visualize eigenvalues/variances(REAL DATA)
fviz_screeplot(res.pca, addlabels = TRUE, ylim = c(0, 63))

#Renamed columns for easy visualization
file.choose()
```

```
Lung_Norm_modR<-read.csv("C:\\Users\\Student175
\\Desktop\\spring 2023\\Thesis\\
1. Dissertation\\Data sets\\Raw data\\
Lung_Norm_modR.csv")
Lung_Norm_modR<-as.matrix(Lung_Norm_modR)
View(Lung_Norm_modR)

##Correlation matrix
Lung_Norm_modR<-cor(Lung_Norm_modR)
View(Lung_Norm_modR)
dim(Lung_Norm_modR)

###PCA
library("factoextra")
library("FactoMineR")

r.pca <- princomp(Lung_Norm_modR)
fviz_eig(r.pca)
fviz_screeplot(r.pca, addlabels = TRUE, ylim = c(0, 63))

fviz_pca_ind(r.pca,
             col.ind = "cos2", # Color by the quality
             of representation gradient.cols =
```

```
             c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
)

fviz_pca_var(r.pca,
             col.var = "contrib", # Color by contributions
             to the PC gradient.cols = c("#00AFBB",
             "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
)

fviz_pca_biplot(r.pca, repel = TRUE,
                col.var = "#2E9FDF", # Variables color
                col.ind = "#696969"  # Individuals color
)

###Loadings

princomp_Lung <- princomp(Lung_Norm_mod)
loadings_Lung <- princomp_Lung$loadings

names(loadings_Lung)
class(loadings_Lung)
```

```
loadings_Lung


#Save the loading as a matrix data


loadings_matrix=loadings( princomp_Lung)[]

View(loadings_matrix)

write.csv(loadings_matrix,"C:/Users/Student175/

Desktop/spring 2023/Thesis/1.Dissertation/

Data sets/Raw data/loadings_matrix.csv")

file.choose()

loadings_matrix<- read.csv("C:\\Users\\

Student175\\Desktop\\spring 2023\\Thesis\\

1. Dissertation\\Data sets\\Raw data\\loadings_matrix.csv")

View(loadings_matrix)

#######


####Summary of loadings for 3 Principal

loadings3_matrix<-loadings_matrix[2:4]

View(loadings3_matrix)

write.csv(loadings3_matrix,"C:/Users/Student175/

Desktop/spring 2023/Thesis/1. Dissertation/

Data sets/Raw data/loadings3_matrix.csv")
```

```
file.choose()
loadings3_matrix<- read.csv("C:\\Users\\
Student175\\Desktop\\spring 2023\\Thesis\\
1. Dissertation\\Data sets\\
Raw data\\loadings3_matrix.csv")
View(loadings3_matrix)

####Column means for PC 1 to 3
colMeans(loadings3_matrix[sapply(loadings3_matrix,
is.numeric)])

####Summary for PC 1 to 3
summary(loadings3_matrix)

#Selection based on loading 1 for variables greater
than 0.1

#####

loading1=loadings3_matrix[,1]

loading1_selection=loading1[which(loading1>=0.1)]
loading1_selection=as.array(loading1_selection)
```

```
match(loading1_selection, loading1)

plot(loading1_selection, pch=19, xlab="axis 1",
ylab="axis 2", main="Selection of loading 1")
#######
#Selection based on loading 2 for variables greater
than 0.1

#####

loading2=loadings3_matrix[,2]

loading2_selection=loading2[which(loading2>=0.1)]

loading2_selection=as.array(loading2_selection)

match(loading2_selection, loading2)

plot(loading2_selection, pch=19, xlab="axis 1",
ylab="axis 2", main="Selection of loading 2")

#######
```

```
#Selection based on loading 3 for variables greater than 0.1

#####

loading3=loadings3_matrix[,3]

loading3_selection=loading3[which(loading3>=0.1)]

loading3_selection=as.array(loading3_selection)

match(loading3_selection, loading3)

plot(loading3_selection, pch=19, xlab="axis 1",
ylab="axis 2", main="Selection of loading 3")

#####a. Heat Map example

# how to make a heatmap in R
library(readxl)
file.choose()
featurematrix<- read_excel("C:\\Users\\Student175
\\Desktop\\jan\\pca\\features selected.xlsx")
View(featurematrix)
```

```
featurematrix<-as.matrix(featurematrix)
heatmap(featurematrix, main = "Test Heat Map")
featurematrix<-cor(featurematrix)
heatmap(featurematrix, main = "Test Heat Map")

##2. Clustering Analysis

file.choose()
LungR_norm<- read.csv("C:\\Users\\Student175\\
Desktop\\jan\\Normalized data\\2. Lung_Norm_mod.csv")
LungR_norm<-as.matrix(LungR_norm)
View(LungR_norm)
LungR_norm<-cor(LungR_norm)
View(LungR_norm)
View(Lung_Norm_mod)

##a. Hierarchical Clustering Analysis

###Calculate distance matrix
distance = dist(LungR_norm)
distance2 = dist(Lung_Norm_mod)

####Hierarchical agglomerative clustering
```

```
##Understanding the Dendogram

mydata.hclust = hclust(distance)

plot(mydata.hclust)

member = cutree(mydata.hclust,3)

table(member)

member


mydata.hclust<-hclust(distance,method="average")

plot(mydata.hclust,hang=-1)

member = cutree(mydata.hclust,3)

table(member)

member


mydata.hclust<-hclust(distance,method="single")

plot(mydata.hclust,hang=-1)

member = cutree(mydata.hclust,3)

table(member)

member


##Variables in each cluster

mydata.hclust<-hclust(distance2,method="average")

plot(mydata.hclust,hang=-1)

member = cutree(mydata.hclust,3)
```

```
table(member)

member


#Silhouette Plot


library(cluster)

plot(silhouette(cutree(mydata.hclust,3), distance))


library(cluster)

plot(silhouette(cutree(mydata.hclust,2), distance))


library(cluster)

plot(silhouette(cutree(mydata.hclust,4), distance))


wss <- (nrow(Lung_Norm_mod)-1)*sum(apply

(Lung_Norm_mod,2,var))for (i in 2:20) wss[i] <-

sum(kmeans(Lung_Norm_mod, centers=i)$withinss)

plot(1:20, wss, type="b", xlab="Number of

Clusters", ylab="Within groups sum of squares")

ot<-LungR_norm


###### Selection of variables per cluster in

Hierachical clustering
```

```
file.choose()
clst1<- read.csv("C:\\Users\\Student175\\Desktop\\
jan\\Clustering\\clst1.csv")
View(clst1)
dim(clst1)
clst2<- read.csv("C:\\Users\\Student175\\Desktop\\
jan\\Clustering\\clst2.csv")
View(clst2)
dim(clst2)
clst3<- read.csv("C:\\Users\\Student175\\Desktop\\
jan\\Clustering\\clst3.csv")
View(clst3)
dim(clst3)

####Summary for clst1
summary(var(clst1))
####Summary for clst2
summary(var(clst2))
####Summary for clst3
summary(var(clst3))

install.packages("reshape2")
library("reshape")
```

```
variancesclst1<-apply(clst1, 2, var)

variancesclst1_Less<-variancesclst1[which
(variancesclst1<=0.02)]

plot(variancesclst1_Less, pch=19, xlab="axis 1",
ylab="axis 2", main="Variance Selection
(Less or Equal 0.02) for Feature from Cluster 1")

write.csv(variancesclst1_Less,"C:/Users/Student182/
Desktop/jan/Clustering/selected/variancesclst1_Less.csv")

##Features in Cluster 2

variancesclst2<-apply(clst2, 2, var)

variancesclst2_Less<-variancesclst2[which
(variancesclst2<=0.02)]

plot(variancesclst2_Less, pch=19, xlab="axis 1",
ylab="axis 2", main="Variance Selection
(Less or Equal 0.02) for Feature from Cluster 2")
write.csv(variancesclst2_Less,"C:/Users/Student175/
```

```
Desktop/jan/Clustering/selected/variancesclst2_Less.csv")

##Features in Cluster 3

variancesclst3<-apply(clst3, 2, var)

variancesclst3_Less<-variancesclst3[which
(variancesclst3<=0.02)]

plot(variancesclst3_Less, pch=19, xlab="axis 1",
ylab="axis 2", main="Variance Selection
(Less or Equal 0.02) for Feature from Cluster 3")

write.csv(variancesclst3_Less,"C:/Users/
Student175/Desktop/jan/Clustering/selected/
variancesclst3_Less.csv")

######### k means
library(factoextra)
library(cluster)

#load data
df <- LungR_norm
```

```
#remove rows with missing values
df <- na.omit(df)

#scale each variable to have a mean of 0 and sd of 1
df <- scale(df)

#Find the Optimal Number of Clusters

#1. Number of Clusters vs. the Total Within
Sum of Squares

#Use the fviz_nbclust() function to create a
plot of the number of clusters vs. the total
within sum of squares:

fviz_nbclust(df, kmeans, method = "wss")

#2. Number of Clusters vs. Gap Statistic
#Calculate the gap statistic for each number
of clusters using the clusGap() function from the
cluster package along with a plot of clusters vs.
gap statistic using the fviz_gap_stat() function:
```

```
#calculate gap statistic based on number of clusters
gap_stat <- clusGap(df,
                    FUN = kmeans,
                    nstart = 25,
                    K.max = 3,
                    B = 50)

#plot number of clusters vs. gap statistic

fviz_gap_stat(gap_stat)

#make this example reproducible
set.seed(1)

#perform k-means clustering with k = 3 clusters
km <- kmeans(LungR_norm, centers = 3, nstart = 25)

#view results
km

#plot results of final k-means model
fviz_cluster(km, data = LungR_norm)
#find means of each cluster
```

```
aggregate(df, by=list(cluster=km$cluster), mean)

#add cluster assignment to original data
final_data <- cbind(df, cluster = km$cluster)

#view final data
head(final_data)

##Each variable clustered
kc<-kmeans(LungR_norm,3)
kc

###Cluster mapping for 3 clusters

datadistshortset<-dist(LungR_norm,method = "euclidean")
hc1 <- hclust(datadistshortset, method = "complete" )
pamvshortset <- pam(datadistshortset,3, diss = FALSE)
clusplot(pamvshortset, shade = FALSE,labels=2,col.clus="blue",
col.p="red",span=FALSE, main="Cluster Mapping",cex=1.2)

pamvshortset_matrix=as.matrix(pamvshortset)

View(pamvshortset_matrix)
```

```
clusters_lung=as.matrix(pamvshortset$clustering)

write.csv(clusters_lung,"C:/Users/Student175/
Desktop/jan/Clustering/clusters_lung.csv")
file.choose()
clusters_lung_matrix<- read.csv("C:\\Users\\Student175
\\Desktop\\jan\\Clustering\\clusters_lung.csv")
View(clusters_lung_matrix)

clusters_lung_matrix<-as.matrix(table
(clusters_lung_matrix))

write.csv(clusters_lung_matrix,"C:/Users/Student175/
Desktop/jan/Clustering/clusters_lung_matrix.csv")
file.choose()
clusters_lung_matrix<- read.csv("C:\\Users\\
Student175\\Desktop\\jan\\
Clustering\\clusters_lung_matrix.csv")
View(clusters_lung_matrix)

###### Selection of variables per cluster k-means
file.choose()
cluster1<- read.csv("C:\\Users\\Student175\\
```

```
Desktop\\jan\\Clustering\\PC1.csv")
View(cluster1)
cluster2<- read.csv("C:\\Users\\Student175\\
Desktop\\jan\\Clustering\\PC2.csv")
View(cluster2)
cluster3<- read.csv("C:\\Users\\Student175\\
Desktop\\jan\\Clustering\\PC3.csv")
View(cluster3)

##Features in Cluster 1

variancesClu1<-apply(cluster1, 2, var)

variancesClu1_Less<-variancesClu1[which
(variancesClu1<=0.02)]

plot(variancesClu1_Less, pch=19, xlab="axis 1",
ylab="axis 2", main="Variance Selection
(Less or Equal 0.02) for Feature from Cluster 1")

write.csv(variancesClu1_Less,"C:/Users/Student175/
Desktop/jan/Clustering/selected/variancesClu1_Less.csv")
##Features in Cluster 2
```

```
variancesClu2<-apply(cluster2, 2, var)

variancesClu2_Less<-variancesClu2[which
(variancesClu2<=0.02)]

plot(variancesClu2_Less, pch=19, xlab="axis 1",
ylab="axis 2", main="Variance Selection
(Less or Equal 0.02) for Feature from Cluster 2")

write.csv(variancesClu2_Less,"C:/Users/Student175/
Desktop/jan/Clustering/selected/variancesClu2_Less.csv")

##Features in Cluster 3

variancesClu3<-apply(cluster3, 2, var)

variancesClu3_Less<-variancesClu3[which
(variancesClu3<=0.02)]

plot(variancesClu3_Less, pch=19, xlab="axis 1",
ylab="axis 2", main="Variance Selection
(Less or Equal 0.02) for Feature from Cluster 3")
write.csv(variancesClu3_Less,"C:/Users/Student175/
```

```
Desktop/jan/Clustering/selected/variancesClu3_Less.csv")


#####ISOMAP

####data

library(readxl)

Lung_CT<- read_excel("C:\\Users\\Student175

\\Desktop\\jan\\Raw data\\Lung_Cancer_CT.xlsx")

LungCT_norm <- as.data.frame(lapply

(Lung_CT, min_max_norm))

LungCT_norm<- read.csv("C:\\Users

\\Student175\\Desktop\\jan\\Raw data\\LungCT_norm.csv")

LungCT_norm<-as.matrix(LungCT_norm)

View(LungCT_norm)

Lung_Norm_mod<-LungCT_norm[ ,

which(apply(LungCT_norm, 2, var) != 0)]

View(Lung_Norm_mod)


####Relevant Packages


install.packages("vegan")


library("vegan")

####ISOMAP logarithm
```

```
head(LungCT_norm[,1:10])

View(LungCT_norm)

dim(LungCT_norm[,1:10])

View(Lung_Norm_mod[,1:10])

dis <- vegdist(Lung_Norm_mod[,1:10])

tr <- spantree(Lung_Norm_mod[,1:10],toolong = 0)

pl <- ordiplot(cmdscale(dis), main="cmdscale")

lines(tr, pl, col="red")

ord <- isomap(dis, k=3)

ord

pl <- plot(ord, main="isomap k=3")

lines(tr, pl, col="red")
```

```
pl <- plot(isomap(dis, k=5), main="isomap k=5")


lines(tr, pl, col="red")


pl <- plot(isomap(dis, epsilon=0.45),

main="isomap epsilon=0.45")


lines(tr, pl, col="red")
```

1.2 SAS Codes

```
<SAS CODE>
/* Generated Code (IMPORT) */
/* Source File: MatrixData_Lung_CT_1_74_Modify.xlsx */
/* Source Path: /home/mostafazahed0/SESUG */
/* Code generated on: 7/17/22, 6:51 PM */
%web_drop_table(WORK.IMPORT);
16
FILENAME REFFILE '/home/mostafazahed0/SESUG/Lung_Cancer_CT.xlsx';
PROC IMPORT DATAFILE=REFFILE
DBMS=XLSX
OUT=Lung;
GETNAMES=YES;
RUN;
PROC CONTENTS DATA=Lung;
RUN;
%web_open_table(WORK.IMPORT);
/*view mean and standard deviation of dataset*/
proc means data=Lung Mean StdDev ndec=3;
run;
/*normalize the dataset*/
proc stdize data=Lung out=normalized_Lung;
 *var values;
```

```
run;
/*print normalized dataset*/
proc print data=normalized_Lung;
/*view mean and standard deviation of normalized dataset*/
proc means data=normalized_Lung Mean StdDev ndec=2;
 *var values;
run;
```

VITA

JANET AKOTH KIRETA

Education:

Bachelor of Economics and Statistics,
South Eastern Kenya University
Kitui, Kenya, May 2015

Master of Science in Mathematical Sciences,
East Tennessee State University,
Johnson City, Tennessee, May 2023

Professional Experience:

Teacher, Obisa Mixed Secondary School,
Oyugis, Kenya, 2012

Teacher, Mithui Mixed Secondary School,
Oyugis, Kenya, 2013–2015

Field Enumerator, OGRA Foundation
Kisumu City, Kenya, 2012–2013

Research Assistant,
Center for Study of Research Adolescence
Kisumu City, Kenya, 2014–2015

Field Officer,
Busara Center for Behavioral Economics
Migori, Kenya, 2015–2016

Research Assistant,
Ideas42 End Line Data Collection Survey
Siaya, Kenya, 2016

Assistant Planning Officer,
National Industrial Training Authority
Nairobi, Kenya, 2016–2021

Graduate Assistant,
East Tennessee State University
Johnson City, Tennessee, 2021–2023