

## **SESUG 2024 Paper 2024**

# **Investigating the Role of Radiomics in PET Image Analysis for Improved T-Stage Tumor Staging**

Victor Aderanti, Mostafa Zahed, East Tennessee State University

## **ABSTRACT**

Early cancer detection is vital for improving patient outcomes worldwide. This study focuses on advancing Region Of Interest (ROI) segmentation and feature extraction in medical imaging through Radiomics techniques, employing tools such as 3D Slicer, Pyradiomics, and Python. Various dimensionality reduction methods, including PCA, K-means, t-SNE, ISOMAP, and Hierarchical Clustering, were utilized to manage high-dimensional features, aiming to improve both interpretability and efficiency. The effectiveness of the reduced feature sets in predicting T-staging, a crucial aspect of the TNM cancer staging system, was evaluated. Multinomial logistic regression models were created and assessed using metrics such as MSE, AIC, BIC, and the Deviance Test. The dataset comprised CT and PET-CT DICOM images from 131 lung cancer patients. Results indicated that PCA identified 14 features, Hierarchical Clustering found 17, t-SNE discovered 58, and ISOMAP identified 40 features, with texture-based features being particularly significant. This research underscores the promise of combining Radiomics with unsupervised learning techniques to enhance cancer prediction capabilities from medical images.

Keywords: Radiomics, PET, Dimension Reduction, Features Extraction, Logit Model

## INTRODUCTION

The primary goal of cancer research is to develop safe and effective methods to prevent, detect, diagnose, treat, and ultimately, cure the various forms of cancer [21]. Cancer research identifies risk factors and their underlying causes, leading to lifestyle and public health recommendations that can reduce cancer incidence [22]. Primary prevention, the ideal goal, seeks to stop cancer from developing in the first place, and this focus on prevention is a testament to the remarkable progress made in our understanding of the disease. Concurrently, cancer treatments and survival rates have improved significantly over the past decades, a testament to the tireless efforts of researchers and clinicians [23]. Continued research can lead to the discovery of new treatment targets, the refinement of surgical techniques, advancements in immunotherapy, and the personalization of medicine [24], all of which hold the potential to transform the landscape of cancer care and improve the lives of countless individuals. As the complexity of cancer becomes increasingly understood, researchers have identified the hallmarks of the disease, providing a roadmap for the development of more effective therapies [21].

## RADIOMICS

Radiomics is a growing field that extracts quantitative features from medical images to characterize tumors and support oncology decisions. It involves four steps: image acquisition, tumor segmentation, feature extraction, and data analysis. This process converts medical images into mineable data, enabling the extraction of hundreds of features that can be used to build predictive models. Radiomics has shown promise in improving treatment response prediction, prognosis, and genetics across various cancer types. However, validation, standardization, and further research are needed to fully implement radiomics in clinical practice. To achieve this, the research aimed to address the following questions;

### Research Question 1

What improvements may be made to ROI segmentation and feature extraction in medical image analysis by combining Radiomics approaches with 3D Slicer, Pyradiomics, and Python?

### Research Question 2

How do dimension reduction approaches (PCA, K-means, t-SNE, ISOMAP, and Hierarchical Clustering) increase high-dimensional feature sets' interpretability and analytical efficiency?

### Research Question 3

Can multinomial logistic regression effectively predict T-stage outcomes with specified features, and how effective are statistical tests (Mean Square Error, AIC, BIC, and Deviance Test) in assessing the model's predictive performance?

## LITERATURE REVIEW

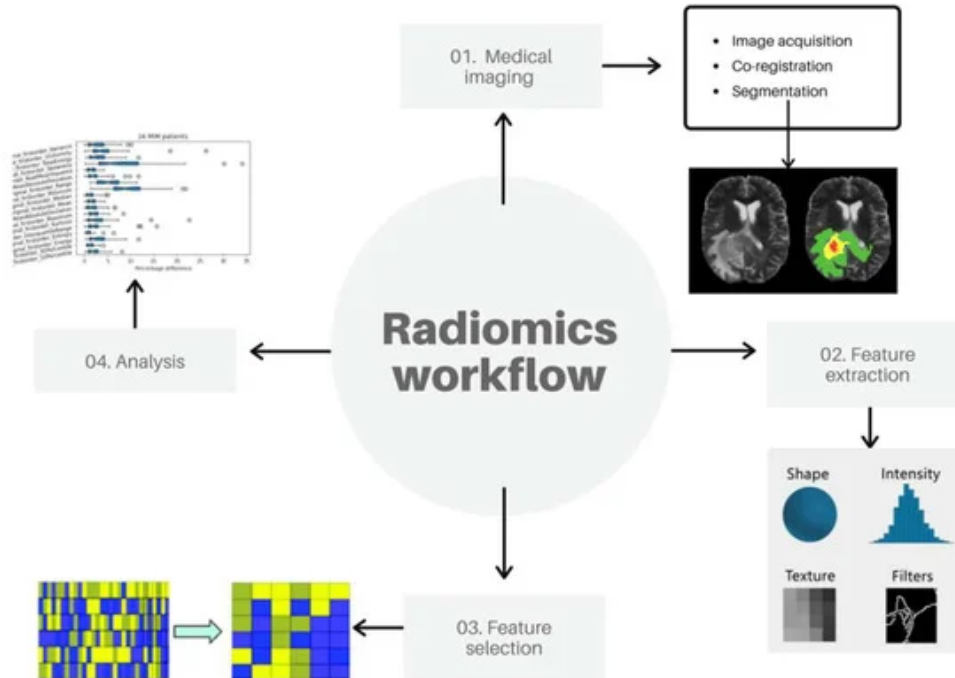
In recent years, there has been a significant surge in research focused on applying radiomics for cancer detection and analysis. The research on using radiomics for cancer detection and analysis has significantly advanced in recent years, demonstrating numerous benefits across various cancer types. Radiomics involves extracting and analyzing quantitative data from medical images to provide insights into tumor characteristics, aiding in prediction, diagnosis, and treatment planning. Studies have shown that radiomic features can reveal tumor heterogeneity, crucial for understanding cancer progression and treatment response. Aerts et al. (2020) found that radiomic features from CT images could predict survival and treatment response in non-small cell lung cancer, outperforming traditional clinical criteria [7]. Parmar et al. (2015) showed that radiomic features from MRI could predict distant metastases in breast cancer, aiding in treatment planning [2]. Bakr et al. (2018) used radiomic features for glioblastoma tumor segmentation in MRI images, achieving higher accuracy than manual segmentation [3]. Dong et al. (2016) demonstrated that radiomic features from MRI could identify breast cancer subtypes based on genetic profiles, offering insights into tumor heterogeneity and therapy responses. Hu et al. (2019) developed a radiomics signature to predict T-stage in lung adenocarcinomas with high predictive performance [4]. Existing research has primarily focused on developing and validating radiomic models for various cancers, exploring the integration with other “omics” technologies, and employing machine learning models for cancer classification. However, gaps remain in optimizing ROI segmentation processes, improving feature extraction techniques, and effectively using unsupervised techniques for deeper insights into tumor heterogeneity. This research aims to address these gaps by integrating advanced radiomics techniques with tools like 3D Slicer, Pyradiomics, and Python, enhancing the accuracy and efficiency of tumor characterization and prediction.

## RESEARCH METHODOLOGY

The research question was examined using a combination of data description and analytical techniques to review and interpret the collected data, drawing meaningful insights.

## DATA DESCRIPTION

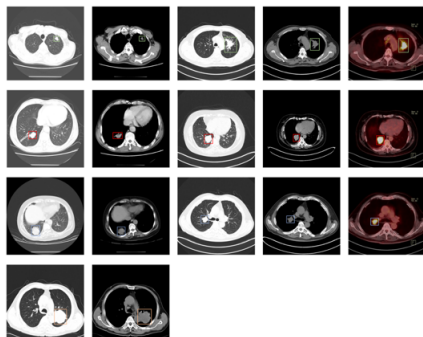
The dataset was gathered by Huiping Han, Funing Yang, and Rui Wang from Harbin Medical University in China [12]. It comprises CT and PET-CT DICOM images of 131 lung cancer patients. The radiomics workflow involves medical imaging, followed by segmentation to define the tumor region. Features are then extracted from this region based on tumor intensity, texture, and shape, and these features are subsequently used for analysis, as illustrated in Figure 1.



**Figure 1. The Radiomics workflow**

Source: <https://www.mdpi.com/2076-3417/11/21/10170>

This dataset consists of PET DICOM images of 131 patient with lung cancer. The XML annotation files, containing the tumor locations, were supplied by five highly experienced academic radiologists specializing in lung cancer to make this dataset a valuable resource for developing medical diagnostic algorithms. Python code run via the terminal was employed to extract the images and enclose them within a bounding box. Pyradiomics, through the 3D Slicer medical software, was used to extract features for 73 out of 131 patients, as the provided annotation file was not compatible with all patients when processed with Python. This dataset is available on The Cancer Imaging Archive (TCIA).



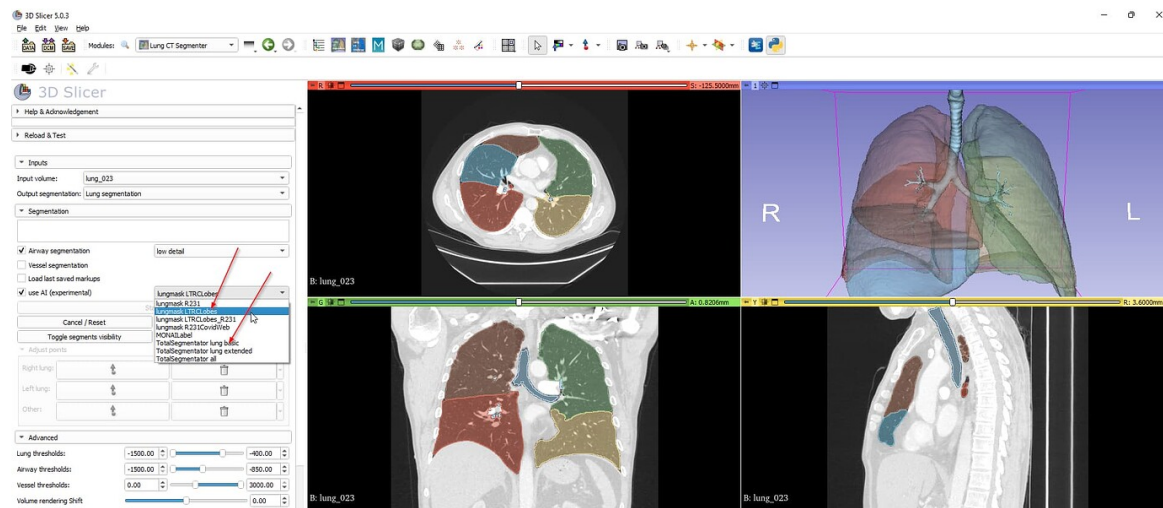
**Figure 2: Visualization of the annotation box on the CT and PET-DICOM images**

Source: <https://wiki.cancerimagingarchive.net>

The process for data acquisition is described as follows:

a) Software tools for extracting features:

There are several software options available for feature extraction, including PyRadiomics, 3D Slicer, LIFEx, IBEX, QIFE, and RayPlus [10]. Each tool has its own set of limitations and restrictions, and no single criterion exists for determining the best one. The choice often depends on the researcher's preferred method for handling medical images. For instance, 3D Slicer is an open-source platform specifically designed for medical image informatics, image processing, and three-dimensional visualization [6]. PyRadiomics is a specialized Python package developed for the extraction of radiomic features [11]. LIFEx is a state-of-the-art software solution focused on extracting quantitative imaging features from various types of medical images. Ultimately, the selection of a software tool depends on the researcher's objectives and level of expertise.

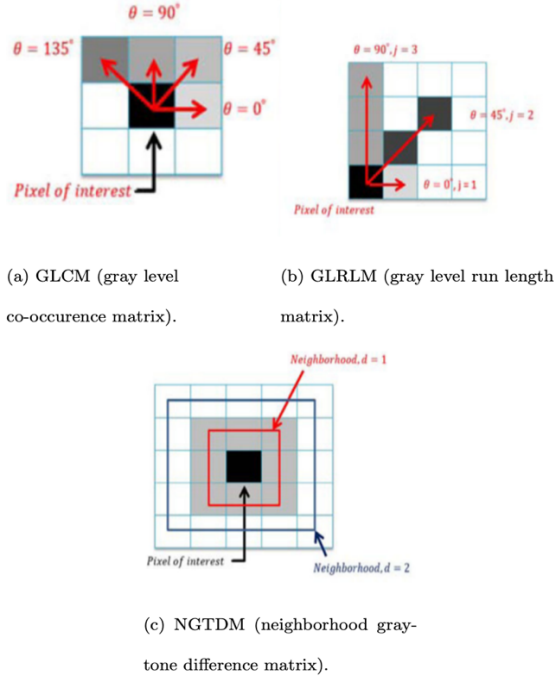


**Figure 3: Loading Lung-CT-PET images**

Source: <https://discourse.slicer.org/t/interactive-lobe-segmentation-problem/26291/2>

b) Extracting features from PET medical images of lung cancer.

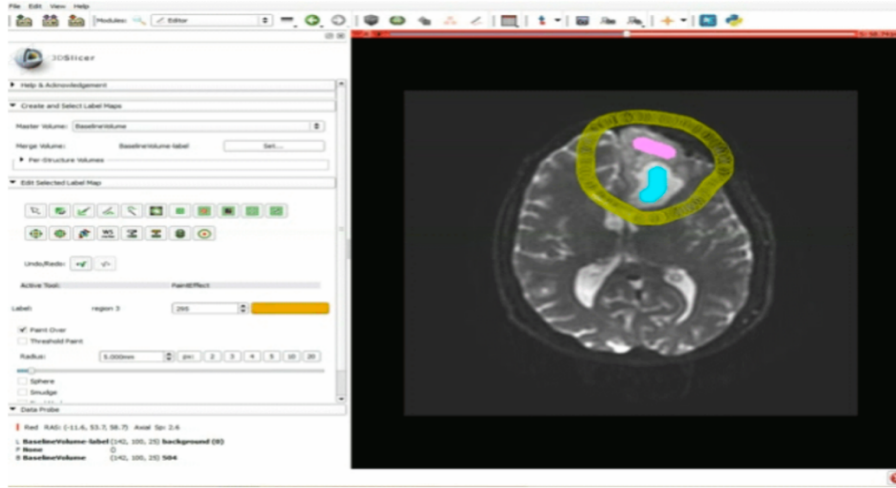
Extracted features are generally classified into three main groups. The first group, first-order radiomics, consists of features based on intensity and shape [9]. The second group, second-order radiomics, includes texture-based features derived from different descriptive matrices, such as the Gray Level Co-occurrence Matrix (GLCM), Gray Level Run Length Matrix (GLRLM), Neighborhood Gray-Tone Difference Matrix (NGTDM), and Gray Level Zone Length Matrix (GLZLM).



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

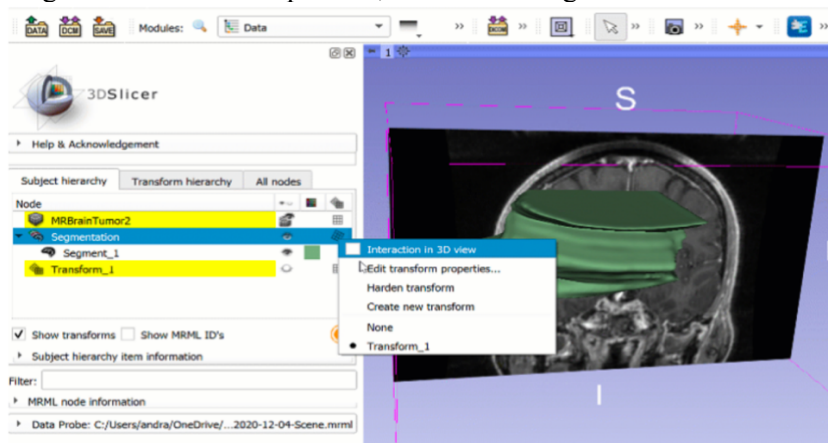
The last category, known as higher order radiomics, employs filters for feature extraction from images. This process involves decomposing tumor images into various frequency domains (such as horizontal, vertical, and diagonal) using wavelet techniques, which enables the extraction of details about tumor shape, intensity, texture, and other attributes. Fourier features are used to gather information on gradients, while Minkowski Functionals (MF) are a common method for higher-order feature extraction, focusing on the arrangement of pixels with intensities above a certain threshold [13].

- c) Extraction process: Out of the 104 patients included in the study, feature extraction was conducted on 98 patients because annotation file issues affected the remaining six. Segmentation was performed using a 3D slicer, with tumors highlighted by a yellow circle and displayed in blue and pink colors, as illustrated in figure 5.



**Figure 5: A tumor being delineated using a 3D slicer**

The PyRadiomics package in the 3D slicer enabled the extraction of tumor features from the segmented images for all evaluated patients, as shown in Figure 6.



**Figure 6: The PyRadiomics package is extracting features.**

#### d) Resulting Data Matrix

Following the feature extraction process, a 73 by 98 unsupervised data matrix was generated. Each row represented a patient, while each column corresponded to an extracted feature from the previously mentioned categories, all being quantitative variables. The data matrix was normalized using min-max normalization to ensure that measurements for each feature were unitless and adaptable to various feature distributions [10]. Due to the matrix's dimensions, traditional reduction techniques like PCA, hierarchical clustering, and k-means encountered difficulties with this format. As a result, the dataset was restructured into a 73 by 68 square correlation matrix, preparing it for effective dimension reduction application.

### Analysis Techniques

In line with the research objective, dimension-reduction techniques were employed to make the data more manageable. These methods included feature extraction and selection [14]. Feature extraction techniques are further divided into supervised and unsupervised learning. Supervised learning considers the

relationship between features and class labels, selecting features based on their ability to differentiate between classes. In contrast, unsupervised learning ignores class labels and focuses on removing redundant features [15]. Given that the data matrix was unsupervised, a linear exploration using Principal Component Analysis (PCA) was conducted to transform the original data into a new set of features that preserved most of the information from the original dataset. Choosing the right dimension-reduction technique depends on the specific dataset and research goals. Applying these techniques enhances computational efficiency, mitigates the curse of dimensionality, and identifies the most important features in the dataset.

## Reduction Technique in Radiomics

### 1. Principal component Analysis (PCA)

Principal Component Analysis (PCA) is a feature transformation technique used to reduce the correlation between sampled variables such as  $x_1, x_2, \dots, x_p$ . Through an orthogonal transformation, PCA creates new variables called principal components  $P_{c1}, P_{c2}, \dots, P_{cm}$  that retain many of the properties of the original variables, with  $m$  being less than  $p$ . This method allows for the generation of new features via linear combinations of the main components, which maximize variance and enhance predictability [25].

The process of performing PCA involves five key steps:

**(a) Data Standardization:** First, calculate the mean of all dimensions of the data set, excluding labels, and scale the data so that each variable contributes equally to the analysis. The formula used for standardization is  $Z = (x - \mu) / \sigma$ , where  $z$  is the scaled value,  $x$  is the original value,  $\mu$  is the mean, and  $\sigma$  is the standard deviation.

**(b) Covariance Matrix Computation:** Identifying highly correlated variables is critical in data analysis because these variables often contain redundant information, which can compromise the accuracy of statistical models. The covariance matrix allows for the examination of correlations between all variable pairs in a dataset, facilitating the removal of redundant variables. The covariance is calculated using  $\text{Cov}(x, y) = (1 / (n-1)) * \sum (x_i - \bar{x})(y_i - \bar{y})$ , where  $\bar{x}$  and  $\bar{y}$  are the means of the predictor and response variables, respectively,  $n$  is the sample size, and  $i$  refers to each observation. However, relying on the covariance matrix alone can cause variables with large variances to dominate the principal components. Additionally, changing measurement units could alter the PCA results, so it is often preferable to base the PCA on the eigenvectors and eigenvalues of the correlation matrix [25].

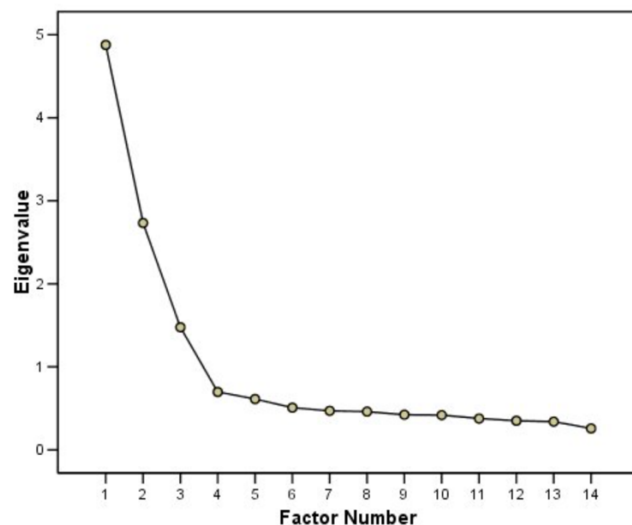
**(c) Eigenvector and Eigenvalue Calculation:** The next step involves using concepts from linear algebra to determine principal components derived from the covariance matrix. An eigenvalue is a scalar that transforms an eigenvector. The relevant equation is  $Av = \lambda v$ , where  $A$  is the square covariance matrix,  $v$  is an eigenvector, and  $\lambda$  is the eigenvalue. Solving this equation gives the eigenvalue  $\lambda$  as  $\det(A - \lambda I) = 0$ , where  $\det$  denotes the determinant,  $A$  is the covariance matrix, and  $\lambda I$  is the scalar multiplying the identity matrix [25].

**(d) Selection of Eigenvectors:** After calculating the eigenvalues, they are sorted in descending order. If the goal is to reduce the dimensionality to two, select the first two eigenvectors corresponding to the two highest eigenvalues.

(e) **Data Reconstruction:** The final step uses the eigenvectors to reorient the data from the original axes to the new ones represented by the principal components. This transformation is given by  $y = W^T * X$ , where  $W^T$  is the transpose of matrix  $W$ ,  $X$  is the eigenvector matrix, and  $y$  is the transformed data. For example, after transformation, the first principal component can be represented as  $Z_1 = a_{11}x_1 + a_{21}x_2 + \dots + a_{p1}x_p$ , where  $a_{11}, a_{21}, \dots, a_{p1}$  are the loadings of the first principal component. Since eigenvalues are the variances of the principal components, we can express the "proportion of variance explained" by the first  $k$  components [10] as:  $(\lambda_1 + \lambda_2 + \dots + \lambda_k) / (\lambda_1 + \lambda_2 + \dots + \lambda_p)$ , where  $\lambda_1$  represents the variance explained by the first component, and so on.

To determine the number of principal components to retain:

- Retain the first  $m$  components that explain a certain percentage (e.g., 70%, 80%, 90%) of the total variance.
- Retain components with eigenvalues greater than or equal to the average eigenvalue or the average sample variance of the original variables.
- Use a scree plot of the eigenvalues, where the number of factors is indicated at the point where the curve flattens (the elbow), see figure 6.

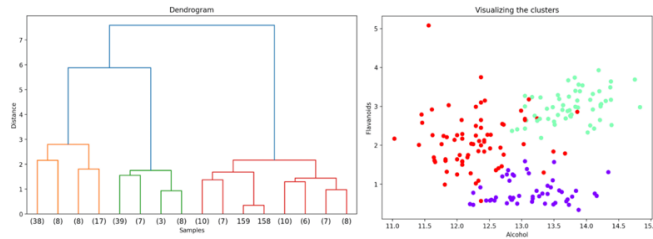


**Figure 6: Scree plot**

**Source:** <https://www.researchgate.net/figure/Graph-depicting-scree-plot-for-selecting-principal-components>.

## 2. Hierarchical Clustering

Agglomerative hierarchical clustering starts with  $n$  clusters, where each cluster contains just one object. In each iteration, the two clusters that are the "nearest" are combined. As the process continues, the number of clusters decreases from  $n$  to  $n-1$ , and so forth. Eventually, in the final stage, there is a single cluster that includes all  $n$  objects.



**Figure 7: Hierarchical Clustering**

Source: <https://www.kdnuggets.com/unveiling-hidden-patterns-an-introduction-to-hierarchical-clustering>.

There are four primary types of linkage methods: complete, average, single (Ward's), and centroid. These linkages are summarized as follows:

- i. **Complete:** In this method, all pairwise dissimilarities between observations within clusters are calculated, and the maximum value is recorded [10].
- ii. **Single (Ward's):** This method computes all pairwise dissimilarities between clusters, recording the minimum value [10].
- iii. **Average:** Here, all pairwise dissimilarities between clusters are computed, and the average value of dissimilarities is recorded [10].
- iv. **Centroid:** In this technique, the dissimilarities between the mean vectors of clusters are calculated, with the mean vector representing the cluster centroid [10].

Steps in the Agglomerative Hierarchical Clustering Algorithm

- a. **Initialization:** Begin with  $N$  clusters, each containing a single entity, and an  $N \times N$  symmetric distance (or similarity) matrix, denoted as  $D = d_{ik}$ .
- b. **Find the Closest Clusters:** Search the distance matrix for the nearest (most similar) pair of clusters. Let the distance between the most similar clusters  $U$  and  $V$  be denoted as  $d_{uv}$ .
- c. **Merge Clusters:** Merge clusters  $U$  and  $V$ , labeling the newly formed cluster as  $UV$ . Update the entries in the distance matrix by [14]:
  - Removing the rows and columns corresponding to clusters  $U$  and  $V$ .
  - Adding a row and column that represent the distances between cluster  $UV$  and the remaining clusters.
- d. **Repeat the Process:** Repeat steps 2 and 3 a total of  $N - 1$  times, ensuring that all objects will be in a single cluster when the algorithm completes. Record the identity of the merged clusters and the levels (distances or similarities) at which these mergers occur.

### 3. t-SNE (t-distributed Stochastic Neighbor Embedding)

This advanced machine learning algorithm is mainly utilized for visualizing high-dimensional data by projecting it onto a lower-dimensional space, usually two or three dimensions. This projection preserves the relative distances between points, thereby maintaining the local structure of the data [17]. The following outlines the steps for applying t-SNE:

### a. Data Preparation

Begin by preprocessing the data, which may include normalization or standardization to ensure that all features contribute equally to the analysis. This step is essential for achieving accurate and meaningful results with t-SNE [16].

### b. Pairwise Distance Calculation

Calculate the pairwise distances between all data points. These distances are crucial for constructing a probability distribution that reflects the similarities between points in the high-dimensional space [17].

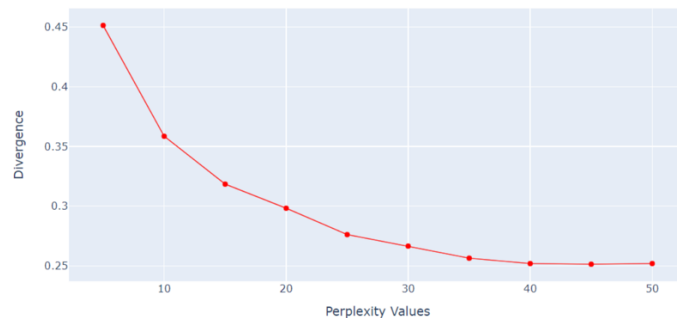
### c. Perplexity and Joint Probability Distribution

The perplexity parameter in t-SNE is a measure of how to balance the attention between local and global aspects of the data. Mathematically, perplexity is defined as:

$$\text{Perplexity } (P_i) = 2^{-\sum_j P(j|i) * \log_2(P(j|i))}$$

where  $P_{ji}$  is the conditional probability that point  $j$  is a neighbor of point  $i$  in the high-dimensional space. A lower perplexity tends to focus on the local structure, while a higher perplexity tends to capture global structure [16].

It's often useful to create a **perplexity plot** to evaluate the effects of different perplexity values on the t-SNE results. The plot typically shows the t-SNE embeddings under different perplexity values, helping to choose an appropriate value for the analysis, see figure 8.



**Figure 8: Perplexity plot**

**Source:** <https://www.datacamp.com/tutorial/introduction-t-sne>

### d. Dimensionality Reduction

Reduce the dimensionality of the data using an iterative optimization algorithm that minimizes the Kullback-Leibler divergence between the original probability distribution and the distribution of the points in the low-dimensional space [16].

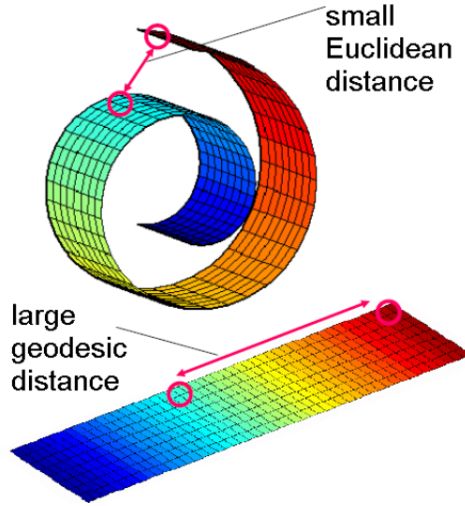
### e. Visualization

Finally, visualize the lower-dimensional representation of the data, which typically reveals clusters or other meaningful patterns that reflect the structure of the original high-dimensional data [17].

## 4. ISOMAP (Isometric Mapping)

This algorithm reduces the dimensionality of data while preserving the distances between data points, with a focus on geodesic distance rather than Euclidean distance. This technique is more effective at capturing

the data's nonlinear structure. The geodesic distance between two data points on a manifold represents the shortest path along the manifold, similar to the great circle distance on a sphere [20], as illustrated in Figure 9.



**Figure 9: A 3D swissroll showing geodesic and euclidean distance**

Source: <https://www.cs.cmu.edu/~bapoczos/Classes/ML10715\ 2015Fall/slides/ManifoldLearning.pdf>.

The ISOMAP algorithm involves the following steps:

a. Construct a neighborhood graph  $G$  from the dataset by connecting only "nearby points" with edges weighted by their Euclidean distances, i.e:

$$d_X(i, j) = \|x_i - x_j\|, \text{ if } x_i \text{ and } x_j \text{ are "close" (otherwise, the distance is 0),}$$

where "closeness" is defined using one of the following methods [20]:

- i.  $\epsilon$ -ball method: A point  $x_j$  is considered close to  $x_i$  if and only if  $\|x_i - x_j\| \leq \epsilon$ .
- ii. k-Nearest Neighbors (kNN) method: A point  $x_j$  is close to  $x_i$  if it is among the  $k$  nearest neighbors of  $x_i$ .

b. Use Dijkstra's algorithm on the nearest neighbor graph  $G$  (created using either method) to calculate the shortest-path distances between all pairs of data points, denoted as  $d_G(i, j)$ .

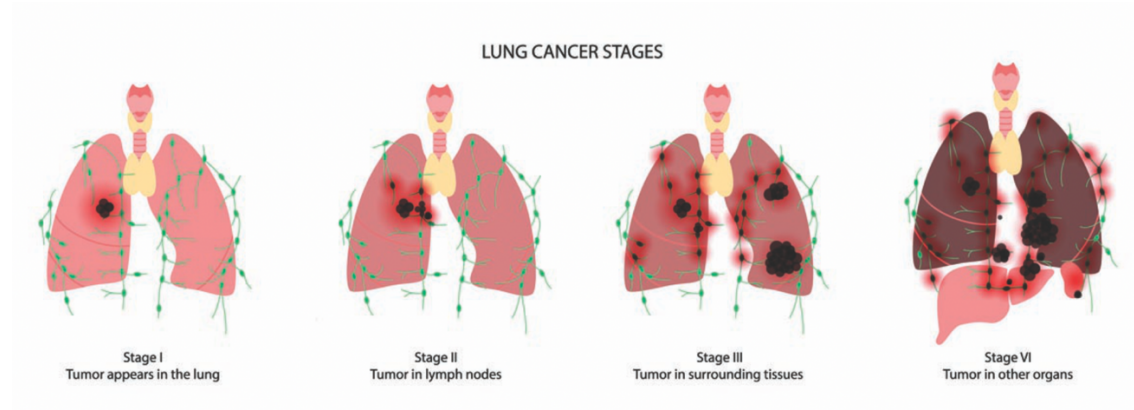
c. Apply Multidimensional Scaling (MDS) using the shortest-path distances  $d_G(i, j)$ . to find a lower-dimensional embedding for the original data [20].

### **T-staging of lung cancer**

Lung cancer, recognized as one of the leading causes of cancer-related mortality worldwide, undergoes assessment and staging predominantly through the TNM classification system [18]. This system delineates the anatomical extent of the disease, with "T" denoting the dimensions and infiltration extent of the primary neoplasm, "N" representing the presence or absence of regional lymphatic dissemination, and "M"

signifying the presence of distant metastases. Notably, the T-stage plays a pivotal role in elucidating the magnitude and dissemination scope of the primary tumor within the pulmonary structures and adjacent anatomical entities. Such delineation is instrumental in guiding therapeutic strategies and prognostic evaluations.

The T-stage of lung cancer is determined through imaging tests such as CT scans, MRI, PET scans, and sometimes surgical reports, providing detailed insights into the tumor's characteristics [19].



**Figure 10: Lung cancer stages**

**Source:** <https://europepmc.org/article/med/29650687>

Figure 10 depicts the different stages of lung cancer, illustrating the development and spread of tumors. In Stage I, a solitary tumor is confined to the lung. By Stage II, the tumor has spread to nearby lymph nodes. Stage III shows multiple tumors present in surrounding tissues, signifying further disease progression. The most severe phase, Stage IV, displays widespread tumor metastasis to other organs, emphasizing the advanced nature of the condition. This visual representation serves as a valuable educational resource for understanding the progression of lung cancer from its early to terminal stages.

## RESULTS FOR ANALYSES

### a. Principal Component Analysis

Utilizing a dataset containing 98 features derived from tumors in PET images of lung cancer patients, we demonstrated the application of dimensionality reduction techniques. This analysis was conducted using SAS software sequentially. The dataset was standardized in SAS, ensuring that each variable had a mean of 0 and a standard deviation of 1. Figure 18 shows a subset of features following the standardization process.

| Means and Standard Deviations from 73 Observations |      |           |
|----------------------------------------------------|------|-----------|
| Variable                                           | Mean | Std Dev   |
| diagnostics_Image.original_Mean_                   | 0.0  | 1.0000000 |
| diagnostics_Image.original_Minim                   | 0.0  | 1.0000000 |
| diagnostics_Image.original_Maxim                   | 0.0  | 1.0000000 |
| diagnostics_Mask.original_VoxelN                   | 0.0  | 1.0000000 |
| diagnostics_Mask.original_Volume                   | 0.0  | 1.0000000 |
| original_shape_Elongation_PET                      | 0.0  | 1.0000000 |
| original_shape_Flatness_PET                        | 0.0  | 1.0000000 |
| original_shape_LeastAxisLength_P                   | 0.0  | 1.0000000 |
| original_shape_MajorAxisLength_P                   | 0.0  | 1.0000000 |
| original_shape_Maximum2DDiameter                   | 0.0  | 1.0000000 |
| original_shape_Maximum2DDiamet_1                   | 0.0  | 1.0000000 |
| original_shape_Maximum2DDiamet_2                   | 0.0  | 1.0000000 |
| original_shape_Maximum3DDiameter                   | 0.0  | 1.0000000 |
| original_shape_MeshVolume_PET                      | 0.0  | 1.0000000 |
| original_shape_MinorAxisLength_P                   | 0.0  | 1.0000000 |
| original_shape_Sphericity_PET                      | 0.0  | 1.0000000 |
| original_shape_SurfaceArea_PET                     | 0.0  | 1.0000000 |
| original_shape_SurfaceVolumeRati                   | 0.0  | 1.0000000 |
| original_shape_VoxelVolume_PET                     | 0.0  | 1.0000000 |
| original_firstorder_10Percentile                   | 0.0  | 1.0000000 |
| original_firstorder_90Percentile                   | 0.0  | 1.0000000 |
| original_firstorder_Energy_PET                     | 0.0  | 1.0000000 |
| original_firstorder_Interquartil                   | 0.0  | 1.0000000 |
| original_firstorder_Kurtosis_PET                   | 0.0  | 1.0000000 |
| original_firstorder_Maximum_PET                    | 0.0  | 1.0000000 |
| original_firstorder_MeanAbsolute                   | 0.0  | 1.0000000 |
| original_firstorder_Mean_PET                       | 0.0  | 1.0000000 |
| original_firstorder_Median_PET                     | 0.0  | 1.0000000 |

**Figure 11: A partial list of features after the standardization process**

The correlation matrix is utilized to calculate the principal components, resulting in a total variance equal to the number of variables, which is 98. SAS software derives the principal components from this correlation matrix; a portion of the correlation matrix is shown in Figure 12.

| The PRINCOMP Procedure |  |              |    |  |  |
|------------------------|--|--------------|----|--|--|
|                        |  | Observations | 73 |  |  |
|                        |  | Variables    | 98 |  |  |

|      | diagnostics_Image.original_Mean_ | diagnostics_Image.original_Minim | diagnostics_Image.original_Maxim | diagnostics_Mask.original_VoxelN | diagnostics_Mask.original_Volume |
|------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|
| Mean | 0.000000000                      | 0.000000000                      | 0.000000000                      | 0.000000000                      | 0.000000000                      |
| Std  | 1.000000000                      | 1.000000000                      | 1.000000000                      | 1.000000000                      | 1.000000000                      |

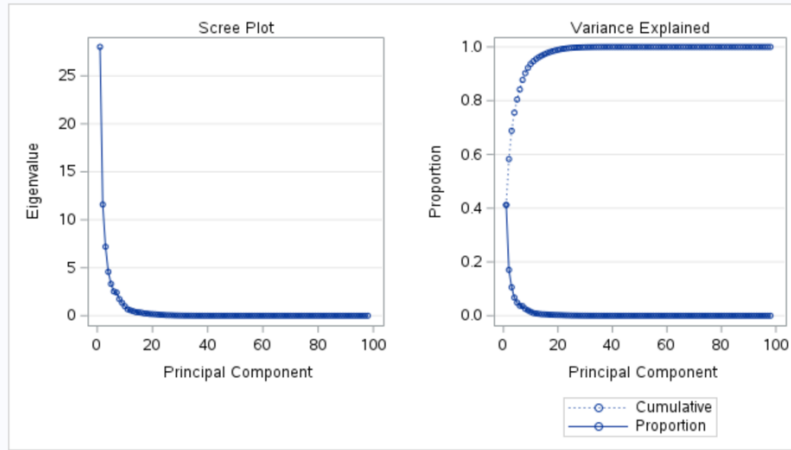
**Figure12: Standardization of the Data**

SAS software calculates the principal components from the correlation matrix. By using eigenvalues to select the principal components, a summary table generated by the software revealed that the first, second, third, fourth, and fifth principal components accounted for approximately 41.21%, 17.05%, 10.57%, 6.73%, and 4.88% of the total variance, respectively. It's important to note that the sum of the eigenvalues equals the total variance. The eigenvalues demonstrated that the first five components effectively summarize the data, capturing 80.44% of the total variance, while each of the remaining components contributes less than 3.5%, as shown in Figure.

| Eigenvalues of the Correlation Matrix |            |            |            |            |
|---------------------------------------|------------|------------|------------|------------|
|                                       | Eigenvalue | Difference | Proportion | Cumulative |
| 1                                     | 28.0194070 | 16.4284635 | 0.4121     | 0.4121     |
| 2                                     | 11.5909435 | 4.4035261  | 0.1705     | 0.5825     |
| 3                                     | 7.1874173  | 2.6140810  | 0.1057     | 0.6882     |
| 4                                     | 4.5733363  | 1.2569529  | 0.0673     | 0.7555     |
| 5                                     | 3.3163834  | 0.7869195  | 0.0488     | 0.8042     |
| 6                                     | 2.5294639  | 0.1170335  | 0.0372     | 0.8414     |
| 7                                     | 2.4124304  | 0.6779596  | 0.0355     | 0.8769     |
| 8                                     | 1.7344709  | 0.3905494  | 0.0255     | 0.9024     |
| 9                                     | 1.3439215  | 0.3631041  | 0.0198     | 0.9222     |
| 10                                    | 0.9808174  | 0.3046413  | 0.0144     | 0.9366     |
| 11                                    | 0.6761761  | 0.1128766  | 0.0099     | 0.9465     |
| 12                                    | 0.5632995  | 0.0951778  | 0.0083     | 0.9548     |
| 13                                    | 0.4681216  | 0.0919816  | 0.0069     | 0.9617     |
| 14                                    | 0.3761400  | 0.0251351  | 0.0055     | 0.9672     |
| 15                                    | 0.3510050  | 0.0365413  | 0.0052     | 0.9724     |
| 16                                    | 0.3144636  | 0.0684190  | 0.0046     | 0.9770     |
| 17                                    | 0.2460446  | 0.0213633  | 0.0036     | 0.9806     |
| 18                                    | 0.2246813  | 0.0427504  | 0.0033     | 0.9839     |
| 19                                    | 0.1819309  | 0.0159813  | 0.0027     | 0.9866     |
| 20                                    | 0.1659496  | 0.0185759  | 0.0024     | 0.9891     |

**Figure 13: Principal Component Analysis of the first 20 features**

A graphical representation was used to determine how many principal components should be retained to summarize the data, depicted as a scree plot of eigenvalues  $\lambda_i$  against factor  $i$ . The plot typically shows a downward curve, with the "elbow" point—where the slope levels off—indicating the number of factors to retain. The first five eigenvalues create a steep curve, followed by a bend and then a shallow slope. The recommendation is to retain the eigenvalues in the steep curve before the straight-line trend begins. The scree plot confirmed that five principal components are sufficient to explain about 80.44% of the variation in the original data. This conclusion is further supported by the variance explained plot, which also shows that five components capture approximately 80.44% of the total variation, as shown in Figure 14.



**Figure14: Scree plot**

To identify which variables contributed most to the principal components, the first principal component (Prin1) was expressed as a linear combination of the original variables based on the eigenvector matrix, as shown in Figure 13.

$$\text{Prin}_1 = 0.029915X_1 + 0.006039X_2 + \dots + 0.00000X_{98}$$

The second principal component of Prin2 was,

$$\text{Prin}_2 = 0.029198X_1 + 0.004183X_2 + \dots + 0.00000X_{98}$$

The second principal component of Prin3 was,

$$\text{Prin}_3 = 0.073959X_1 + 0.180865X_2 + \dots + 0.00000X_{98}$$

The second principal component of Prin3 was,

$$\text{Prin}_4 = 0.157262X_1 + 0.143096 X_2 + \dots + 0.00000X_{98}$$

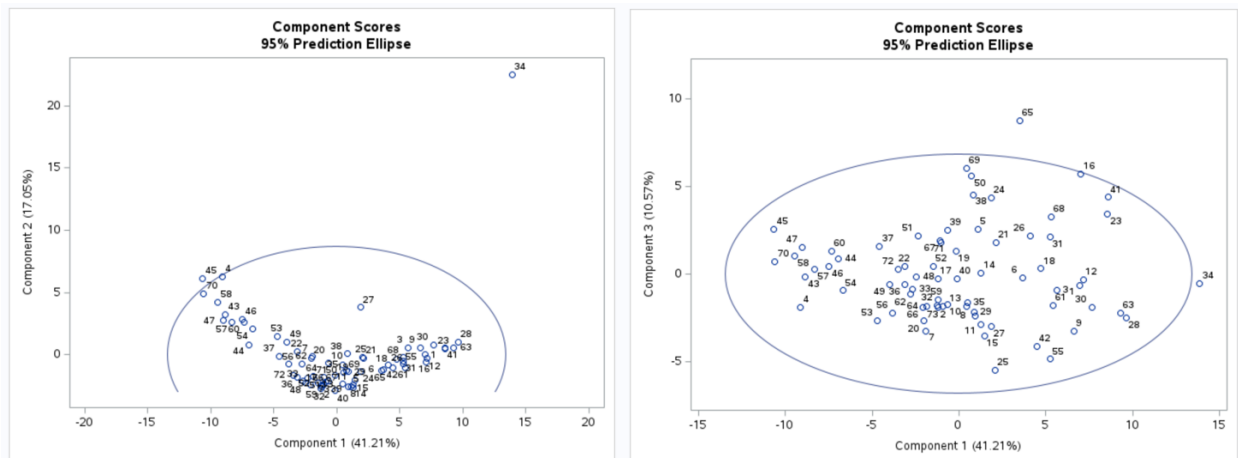
The second principal component of Prin5 was,

$$\text{Prin}_5 = -0.080680X_1 + 0.110461X_2 - \dots - 0.00000X_{98}$$

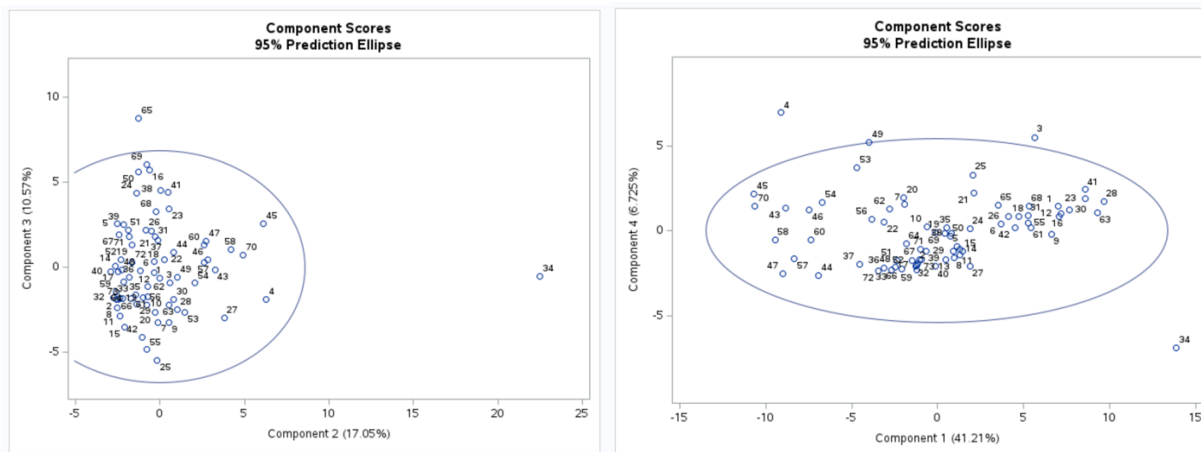
|                                    |                                            | Prin1     | Prin2     | Prin3     | Prin4     | Prin5     | Prin6     | Prin7     | Prin8     | Prin9     | Prin10    |
|------------------------------------|--------------------------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| diagnostics_Image.original_Mean_   | diagnostics_Image.original_Mean_PET        | 0.029915  | 0.006039  | -1.55866  | -1.15581  | 0.145578  | 0.143339  | 0.153584  | -1.71924  | 0.477671  | -0.009559 |
| diagnostics_Image.original_Minim   | diagnostics_Image.original_Minimum_PET     | 0.029198  | 0.004183  | -0.71976  | 0.013272  | -0.13212  | 0.172408  | 0.162950  | -0.221503 | 0.650583  | 0.145597  |
| diagnostics_Image.original_Maxim   | diagnostics_Image.original_Maximum_PET     | 0.073959  | 0.013786  | 0.180865  | 0.075345  | -1.101882 | -1.15331  | -0.08957  | 0.189716  | 0.123389  | -0.423927 |
| diagnostics_Mask.original_VoxelN   | diagnostics_Mask.original_VoxelNum_PET     | 0.157262  | 0.143096  | -0.80358  | -0.016660 | -0.061708 | -0.004996 | 0.039747  | -0.019853 | -0.05211  | -0.005166 |
| diagnostics_Mask.original_Volume   | diagnostics_Mask.original_VolumeNum_PET    | -0.080680 | 0.110461  | -0.069250 | 0.273137  | 0.153755  | -0.026965 | -0.067001 | 0.274051  | 0.083713  | -0.013016 |
| original_shape_Elongation_PET      | original_shape_Elongation_PET              | 0.050932  | -0.010304 | 0.065245  | 0.005780  | -1.09575  | -2.377743 | 0.368657  | 0.190082  | 0.165197  | -0.256832 |
| original_shape_Flatness_PET        | original_shape_Flatness_PET                | 0.058830  | 0.232829  | -0.022686 | -1.86465  | 0.150083  | -1.04861  | -0.016721 | 0.000021  | -0.033662 | -0.03790  |
| original_shape_LeastAxisLength_P   | original_shape_LeastAxisLength_PET         | 0.053156  | 0.212810  | -0.035582 | -1.77686  | 0.178578  | -0.065456 | -1.02566  | -0.035955 | -0.039592 | -0.005510 |
| original_shape_MajorAxisLength_P   | original_shape_MajorAxisLength_PET         | 0.096678  | 0.036912  | -1.19662  | 0.022393  | 0.158560  | 0.218744  | -0.366448 | -1.11952  | -0.047510 | 0.015350  |
| original_shape_Maximum2DDiameter   | original_shape_Maximum2DDiameterColumn_PET | 0.123054  | 0.048617  | -1.09481  | 0.047970  | 0.089701  | 0.180950  | -0.247089 | -1.04443  | -0.034051 | -0.079832 |
| original_shape_Maximum2DDiameter_1 | original_shape_Maximum2DDiameterRow_PET    | 0.153786  | 0.059493  | -0.84127  | 0.053085  | 0.016200  | 0.016429  | -1.05677  | -0.072552 | 0.062096  | -0.057875 |
| original_shape_Maximum2DDiameter_2 | original_shape_Maximum2DDiameterSlice_PET  | 0.124671  | 0.042779  | -1.32766  | 0.144968  | 0.069250  | 0.129950  | -0.225513 | 0.045136  | 0.032244  | 0.016000  |
| original_shape_Maximum3DDiameter   | original_shape_Maximum3DDiameter_PET       | 0.121493  | 0.059252  | -1.40427  | 0.114422  | 0.115603  | 0.141707  | -0.294033 | 0.009248  | 0.019432  | 0.012851  |
| original_shape_MeshVolume_PET      | original_shape_MeshVolume_PET              | 0.156612  | 0.137200  | -0.084839 | -0.011150 | -0.079066 | 0.003158  | 0.049333  | -0.003984 | -0.040601 | 0.010168  |
| original_shape_MinorAxisLength_P   | original_shape_MinorAxisLength_PET         | 0.170442  | 0.058632  | -0.086313 | 0.067424  | -0.043080 | 0.010350  | 0.019381  | 0.012971  | 0.060676  | -1.05865  |
| original_shape_Sphericity_PET      | original_shape_Sphericity_PET              | -1.68618  | 0.036220  | 0.088984  | -0.061583 | -1.05353  | 0.057343  | 0.036395  | -0.034503 | -0.060259 | 0.010167  |
| original_shape_SurfaceArea_PET     | original_shape_SurfaceArea_PET             | 0.157185  | 0.141779  | -0.083493 | -0.015062 | -0.063601 | -0.005386 | 0.039934  | -0.014161 | -0.006166 | -0.008601 |
| original_shape_SurfaceVolumeRati   | original_shape_SurfaceVolumeRatio_PET      | -1.52677  | 0.093910  | 0.068506  | 0.004481  | -0.094614 | 0.104455  | -0.058581 | -0.079805 | 0.079271  | -0.098295 |
| original_shape_VoxelVolume_PET     | original_shape_VoxelVolume_PET             | 0.156664  | 0.137385  | -0.085209 | -0.010808 | -0.077575 | 0.002839  | 0.047453  | -0.004921 | -0.038414 | 0.010319  |
| original_firstorder_10Percentile   | original_firstorder_10Percentile_PET       | 0.056014  | -0.017579 | 0.146370  | 0.022525  | 0.239472  | 0.390629  | 0.227344  | 0.099874  | -0.081345 | -0.065877 |
| original_firstorder_90Percentile   | original_firstorder_90Percentile_PET       | 0.124721  | 0.023160  | 0.262965  | 0.090146  | 0.038991  | 0.016260  | -0.031670 | -0.038756 | 0.030808  | 0.045102  |
| original_firstorder_Energy_PET     | original_firstorder_Energy_PET             | 0.124822  | 0.174997  | 0.092586  | -0.038067 | 0.049496  | 0.028010  | 0.119760  | 0.035513  | -0.034866 | -0.037698 |
| original_firstorder_Interquartil   | original_firstorder_InterquartileRange_PET | 0.113461  | 0.037044  | 0.249202  | 0.083958  | -0.028377 | -1.44824  | -1.281156 | -0.081481 | 0.033040  | 0.086988  |
| original_firstorder_Kurtosis_PET   | original_firstorder_Kurtosis_PET           | -0.151608 | 0.010552  | -1.97768  | 0.107365  | -0.026135 | 0.015677  | -0.007538 | 0.433674  | 0.179729  | -0.263547 |

**Figure 15: Feature Loadings for 10 Principal Components**

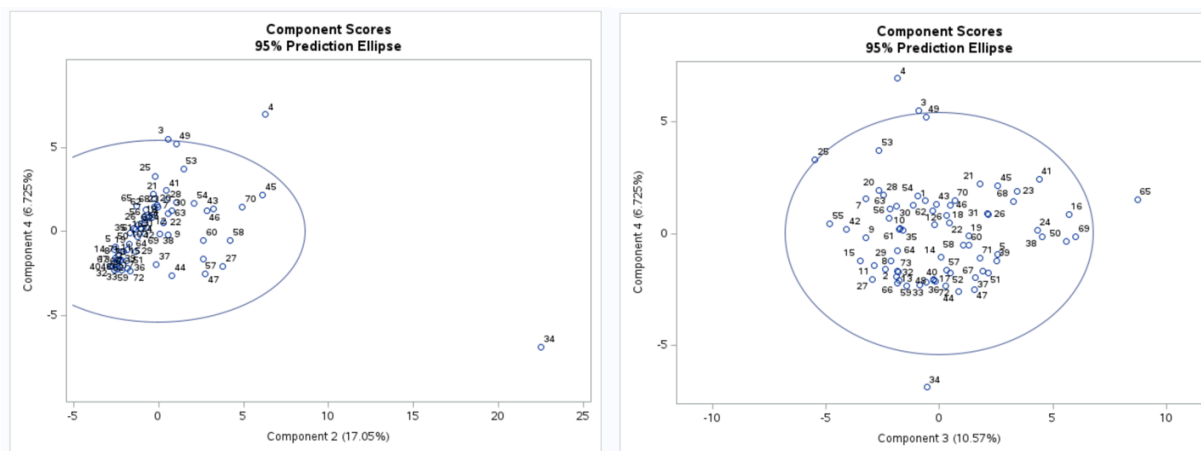
Also, we can illustrate the pairwise component score plots for the first components, with a 95% prediction ellipse overlaid on each scatter plot, Figures, 16,17,18, and 19. Figures 16 and 18 show the plot of the first components. The plots indicate regional trends in the plot of the first two components. Assuming components 2 and 3 are from a bivariate normal distribution, the ellipse identifies extracted features 26.



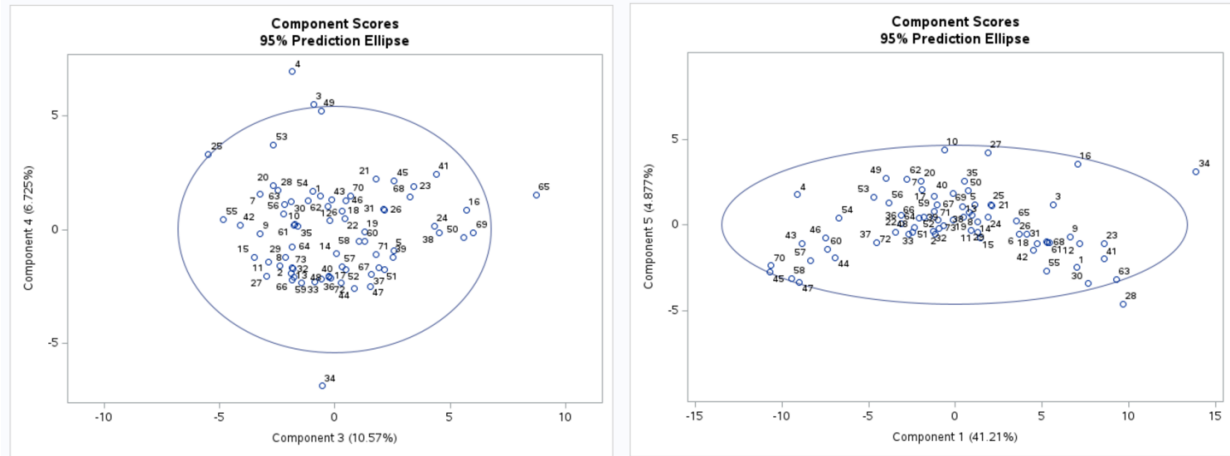
**Figure 16: Plot of the First Three Component Scores**



**Figure 17: Plot of the Third and Fourth Component Scores**



**Figure 18: Plot of the Fourth Component Scores**



**Figure 19: Plot of the Fourth and Fifth Component Scores**

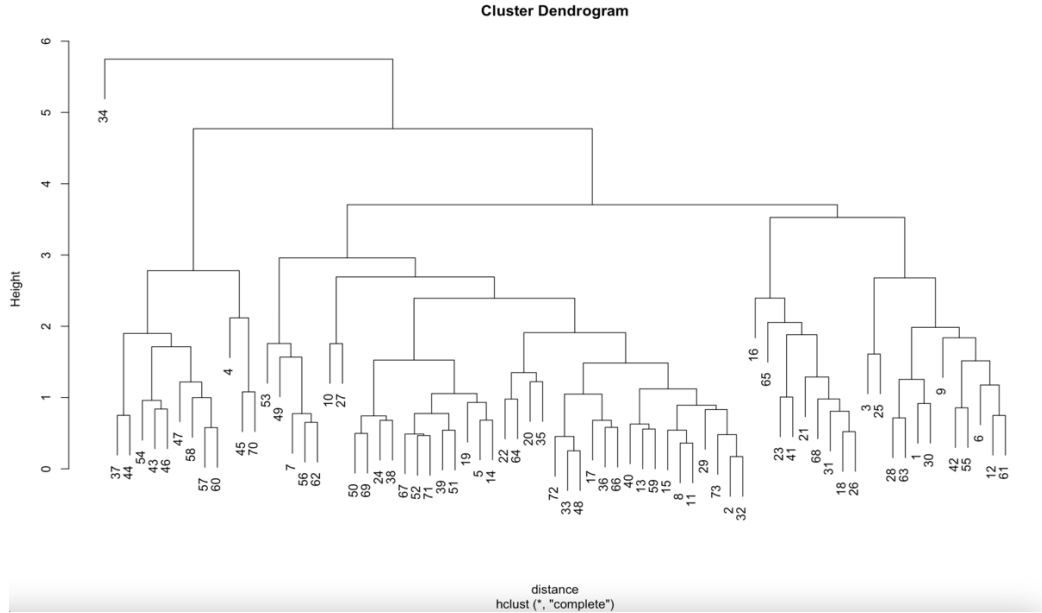
Since the main objective of the research was to reduce the number of variables, a low loadings filter was applied using five selected principal components. A threshold value of 0.15 was established based on the five-number summary for each variable within these components, leading to the removal of features with loadings below this threshold. As a result, 26 features were retained from the original dataset of 98. A summary of selected PCA features, as shown in Table 1.

| Reduction Technique                | Principal Component | Categories | Number of Variable Selected | Percentage |
|------------------------------------|---------------------|------------|-----------------------------|------------|
| Principal Component Analysis (PCA) | PC1                 | Texture    | 10                          | 38.5       |
|                                    |                     | Shape      | 1                           | 3.8        |
|                                    |                     | Intensity  | 3                           | 11.5       |
|                                    | PC2                 | Texture    | 1                           | 3.8        |
|                                    |                     | Shape      | 0                           | 0          |
|                                    |                     | Intensity  | 0                           | 0          |
|                                    | PC3                 | Texture    | 0                           | 0          |
|                                    |                     | Shape      | 2                           | 7.7        |
|                                    |                     | Intensity  | 0                           | 0          |
|                                    | PC4                 | Texture    | 5                           | 19.2       |
|                                    |                     | Shape      | 0                           | 0          |
|                                    |                     | Intensity  | 0                           | 0          |
|                                    | PC5                 | Texture    | 0                           | 0          |
|                                    |                     | Shape      | 4                           | 15.4       |
|                                    |                     | Intensity  | 0                           | 0          |
|                                    |                     |            | 26                          | 100        |

**Table 1: Selected Features in Principal component analysis**

### a. Hierarchical Clustering

The agglomerative hierarchical technique was applied to the lung cancer data characteristics using complete, average, and Ward's minimum-variance clustering methods, all conducted in the R programming language. A dendrogram, which visually represents the cluster hierarchy, shows the distances between clusters and the order in which they merge. The branch height in the dendrogram indicates the distance between clusters, while the proximity of branches reflects the similarity between them. Visual representations can be seen in Figures 20.



**Figure 20: Cluster Dendrogram**

The analysis proceeded by applying a variance filter to each cluster to identify the most similar features within them. A five-number summary of the selected feature matrix determined that a 25th percentile average of 0.01 was an appropriate threshold, leading to the exclusion of features with a variance greater than 0.05. The final selection included 17 features: 5 in Cluster 1, 6 in Cluster 2, 2 in Cluster 3, and 4 in Cluster 4. In Cluster 1, there were 2 intensity-based, 1 shape-based, and 2 texture-based features. Cluster 2 contained 6 features, consisting of 1 intensity-based, 1 shape-based, and 4 texture-based features. Cluster 3 had 2 texture-based features, with no shape or intensity features, while Cluster 4 included 1 texture-based and 3 intensity-based features, with no shape features. The findings concluded that texture-based features were the most dominant, contributing the largest percentage to the new data matrix. A summary of these findings is presented in Table 3.

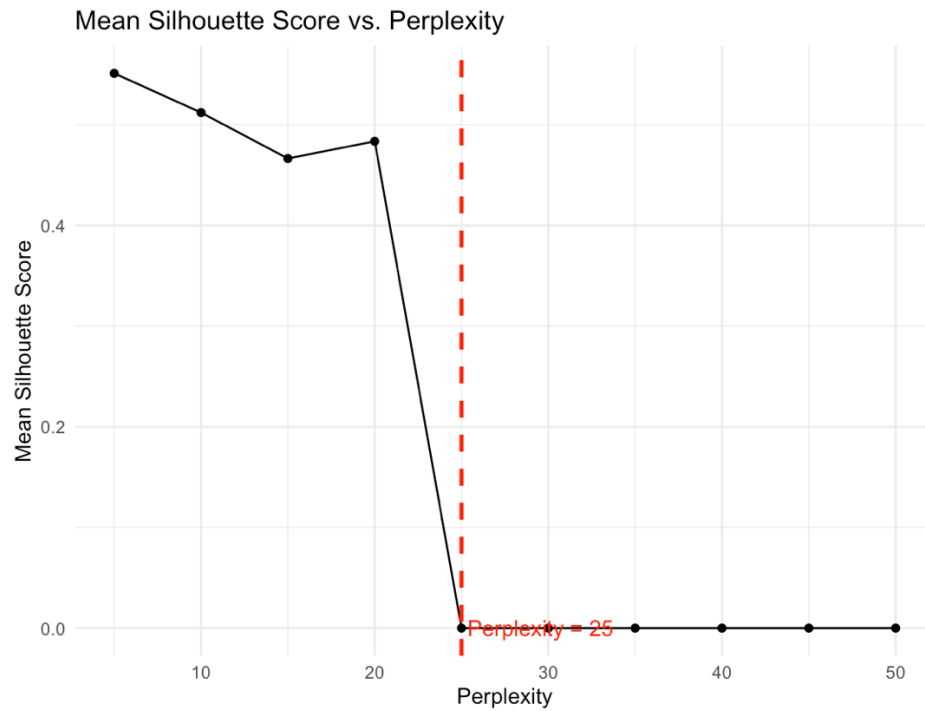
| Reduction Technique     | Principal Component | Categories | Number of Variable Selected | Percentage |
|-------------------------|---------------------|------------|-----------------------------|------------|
| Hierarchical Clustering | Cluster 1           | Texture    | 2                           | 11.8       |
|                         |                     | Shape      | 1                           | 5.9        |
|                         |                     | Intensity  | 2                           | 11.8       |
|                         | Cluster 2           | Texture    | 4                           | 23.5       |
|                         |                     | Shape      | 1                           | 5.9        |
|                         |                     | Intensity  | 1                           | 5.9        |
|                         | Cluster 3           | Texture    | 2                           | 11.8       |
|                         |                     | Shape      | 0                           | 0          |
|                         |                     | Intensity  | 0                           | 0          |
|                         | Cluster 4           | Texture    | 1                           | 5.9        |
|                         |                     | Shape      | 0                           | 0          |
|                         |                     | Intensity  | 3                           | 17.6       |
|                         |                     | Total      | 17                          | 100        |

**Table 3: Selected Features in Hierarchical Clustering**

b. t-SNE (distributed stochastic neighbor embedding)

This technique is particularly well-suited for the visualization of high-dimensional datasets and is widely used in machine learning for data exploration. The perplexity parameter in t-SNE represents the effective number of neighbors for each point, similar to the nearest neighbors concept in other manifold learning techniques. This parameter significantly influences the t-SNE embedding and is typically determined through experimentation. It can be seen as an estimate of how many nearby neighbors each point has, with values in this context ranging from close to 0 up to 50.

Figure 21 shows that the highest mean silhouette scores occur at lower perplexity levels, particularly around 5. As the perplexity increases, the silhouette score drops sharply before stabilizing at a point marked by a vertical red dashed line, suggesting that a perplexity value around 25 may be optimal for this dataset. This value avoids the extremes of too few neighbors, which could misrepresent local data structures, and too many neighbors, which could blur distinct clusters. The red dashed line at perplexity 25 indicates the balance point or the optimal perplexity, identified using the elbow method where further increases no longer significantly improve the silhouette score. The plot is designed to help select a perplexity value that balances capturing both local and global properties of the dataset, aiming for clear, well-separated clusters.



**Figure 21: Perplexity plot**

The analysis advanced by applying a variance filter to each cluster, aiming to pinpoint features within each cluster that exhibited the most similarity. A five-number summary of the selected feature matrix revealed that an average 25th percentile value of 0.01 was an appropriate threshold, leading to the exclusion of any features with a variance exceeding 0.05. The final selection comprised 57 features: 5 from Cluster 1, 6 from Cluster 2, 2 from Cluster 3, and 4 from Cluster 4. Cluster 1's features included 2 intensity-based, 1 shape-based, and 2 texture-based features. Cluster 2 contained 6 features, consisting of 1 intensity-based, 1 shape-based, and 4 texture-based features. Cluster 3 had 2 texture-based features, with no features related to shape or intensity, while Cluster 4 included 1 texture-based and 3 intensity-based features, with none related to shape. The analysis concluded that texture-based features were the most prevalent, contributing the largest percentage to the new data matrix. Table 4 summarize the findings, listing the selected features for each component.

| Reduction Technique                                | Clusters  | Categories | Number of Variable Selected | Percentage |
|----------------------------------------------------|-----------|------------|-----------------------------|------------|
| t-SNE(t-distributed stochastic neighbor embedding) | Cluster 1 | Texture    | 10                          | 17.2       |
|                                                    |           | Shape      | 0                           | 0          |
|                                                    |           | Intensity  | 2                           | 3.4        |
|                                                    | Cluster 2 | Texture    | 15                          | 25.9       |
|                                                    |           | Shape      | 2                           | 3.4        |
|                                                    |           | Intensity  | 9                           | 15.5       |
|                                                    | Cluster 3 | Texture    | 1                           | 1.7        |
|                                                    |           | Shape      | 0                           | 0          |
|                                                    |           | Intensity  | 6                           | 10.3       |
|                                                    | Cluster 4 | Texture    | 5                           | 8.6        |
|                                                    |           | Shape      | 6                           | 10.3       |
|                                                    |           | Intensity  | 1                           | 1.7        |
|                                                    | Cluster 5 | Texture    | 0                           | 0          |
|                                                    |           | Shape      | 1                           | 1.7        |
|                                                    |           | Intensity  | 0                           | 0          |
| Total                                              |           |            | 58                          | 100        |

**Table 4: Selected Features in t-SNE**

c. ISOMAP (Isometric Mapping)

The research aimed to project high-dimensional data onto a lower-dimensional space while preserving the underlying structure using a non-linear dimensionality reduction technique. To reduce the number of variables, a low loadings filter was applied across five selected components. A threshold of 1.5, determined from the five-number summary for each variable within the components, was used to exclude features with loadings below this value. This process resulted in the retention of 40 features from the original 68 in the normalized dataset. The selected features were distributed as follows: Component One included 20 features (2 intensity-based, 2 shape-based, and 16 texture-based), Component Two had 9 features (7 texture-based and 2 shape-based), Component Three consisted of 10 shape-based features, and Component Four contained 1 texture-based feature. The findings highlighted that texture features were the most significant according to the ISOMAP technique. A summary of the selected ISOMAP features is provided, with Table 4 displaying the features selected in each component.

| Reduction Technique           | Selected features | Categories | Number of Variable Selected | Percentage |
|-------------------------------|-------------------|------------|-----------------------------|------------|
| ISOMAP<br>(Isometric Mapping) | Component 1       | Texture    | 16                          | 40         |
|                               |                   | Shape      | 2                           | 5          |
|                               |                   | Intensity  | 2                           | 5          |
|                               | Component 2       | Texture    | 7                           | 17.5       |
|                               |                   | Shape      | 2                           | 5          |
|                               |                   | Intensity  | 0                           | 0          |
|                               | Component 3       | Texture    | 0                           | 0          |
|                               |                   | Shape      | 10                          | 25         |
|                               |                   | Intensity  | 0                           | 0          |
|                               | Component 4       | Texture    | 1                           | 2.5        |
|                               |                   | Shape      | 0                           | 0          |
|                               |                   | Intensity  | 0                           | 0          |
|                               |                   | Total      | <b>40</b>                   | <b>100</b> |

**Table 4: Selected Features in Isometric Mapping**

### Multinomial Logistic Regression Model

It estimates the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables [41]. The formula for the Cumulative Logit Model is:

$$\text{Logit} \left( \frac{P_{\leq j}}{P_{\leq j}} \right) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}$$

where P is the probability of falling in the T-stage category, j is 1, 2, 3, and 4. Note:  $X_1, X_2, \dots, X_p$  are extracted features. We employed a multinomial logistic regression model, using the logit approach, on the features selected through PCA, hierarchical clustering, t-SNE, and isometric mapping. In evaluating the multinomial logistic regression model, we will apply several statistical tests and metrics commonly used for model assessment and hypothesis testing. The following methods and tools will be utilized:

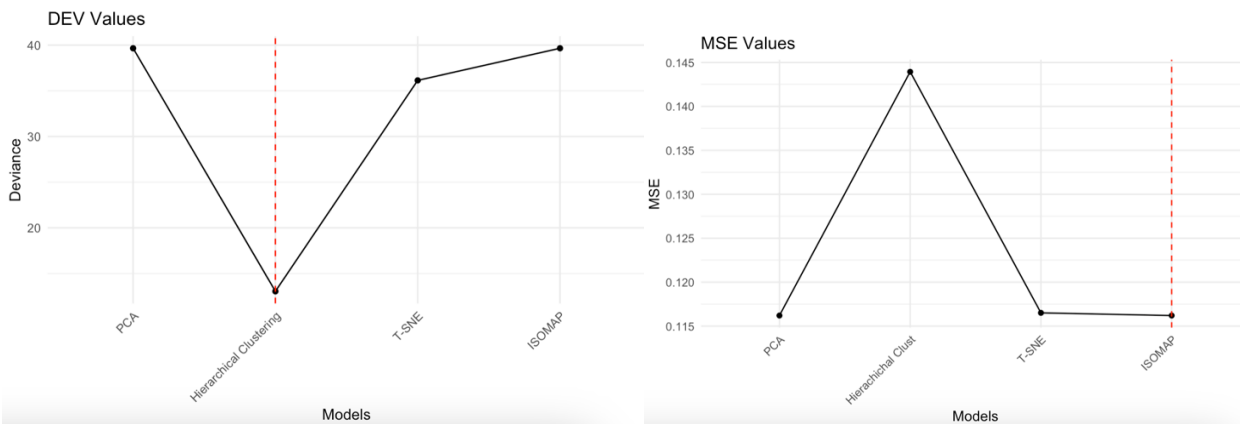
- AIC and BIC: Information criteria that balance model fit and complexity, with lower values indicating a better-fitting model [17].
- DEV (Deviance statistics): Compares the fit of the full model to a reduced model with fewer predictors. It helps assess whether the inclusion of specific predictors improves the model fit.
- MSE (Mean Square Error): It evaluates the level of error in statistical models by measuring the average squared difference between the observed and predicted values.

Figure 22 displays AIC values for different models, with t-SNE having the lowest AIC, making it the preferred model according to this metric. A red dashed line highlights t-SNE as the top model. Similarly, the BIC values are shown for various models, with Hierarchical Clustering having the lowest BIC, marking it as the best choice. The red dashed line in the graph emphasizes Hierarchical Clustering as the leading model.



**Figure 22: AIC and BIC value of the model**

Figure 23 shows the Mean Squared Error (MSE) values for the cumulative logit model applied to PCA, Hierarchical Clustering, t-SNE, and ISOMAP. ISOMAP has the lowest MSE, indicating it offers the most accurate predictions. The figure also displays deviance (DEV) values, with Hierarchical Clustering showing the lowest deviance, suggesting it provides the best model fit among the techniques.



**Figure 23: DEV and MSE value of the model**

Figure 24 presents a summary of the cumulative logit model applied to the PCA, Hierarchical Clustering, t-SNE, and ISOMAP techniques. Lower AIC and BIC values indicate a better model fit with fewer parameters, while lower deviance and MSE values reflect higher model accuracy.

t-SNE has the lowest AIC (172.3715), indicating the best fit with fewer parameters. Hierarchical Clustering shows the lowest BIC (216.1066) and deviance (13.0517), suggesting a strong overall fit. Both ISOMAP and t-SNE demonstrate the lowest MSE values (0.116202 and 0.116508), indicating that they are the most accurate techniques for dimensionality reduction in this model.

| CUMMULATIVE LOGIT MODEL | AIC      | BIC      | DEV     | MSE       |
|-------------------------|----------|----------|---------|-----------|
| PCA                     | 174.857  | 220.6661 | 27.8512 | 0.1311304 |
| HIERRACHICAL CLUSTERING | 179.4592 | 216.1066 | 13.0517 | 0.1439367 |
| T-SNE                   | 172.3715 | 227.3426 | 36.1394 | 0.116508  |
| ISOMAP                  | 176.8527 | 240.9855 | 39.6582 | 0.116202  |

**Figure 24: Cumulative logit model**

## CONCLUSION

The research primarily aimed to determine if the initial research questions were effectively addressed by reducing the original set of 98 variables. The study successfully identified a smaller set of features, selecting 26 features for PCA, 17 for Hierarchical Clustering, 58 for t-SNE, and 40 for ISOMAP. The analyses consistently highlighted texture-based features as the most significant category. The dimensionality reduction achieved was substantial, with the feature sets reduced by approximately 75%, resulting in smaller data matrices (e.g., 73 by 26 for PCA). Specifically, for PCA, five main components were identified as significant, capturing 82.7% of the variance. The analysis identified 26 significant features using a 0.15 loadings filter, with texture-based features being the most dominant across all principal components. In the clustering analysis, 17 features were selected using a variance filter with a 0.05 cutoff. Texture-based features again emerged as the most dominant in the new data matrix. Similarly, t-SNE and ISOMAP analyses reinforced the significance of texture-based features. Finally, Multinomial Logistic Regression (MLR) was applied to the selected features from each method (PCA, Hierarchical Clustering, t-SNE, and ISOMAP) to assess T-stage as an outcome using the cumulative logit model. The analysis provided a comprehensive evaluation of the predictive performance and significance of the model, further confirming the critical role of texture-based features in the dataset.

## REFERENCE

1. Hosmer, D.W., Lemeshow, S., & Sturdivant, R.X. (2013). "Applied Logistic Regression" (3rd ed.). Wiley Series in Probability and Statistics.
2. Parmar, C., Grossmann, P., Rietveld, D., Rietbergen, M. M., Lambin, P., & Aerts, H. J. (2015). Radiomic machine-learning classifiers for prognostic biomarkers of head and neck cancer. *Frontiers in Oncology*, 5, 272. <https://doi.org/10.3389/fonc.2015.00272>.

3. Bakr, S., Gevaert, O., Echegaray, S., Ayers, K., Zhou, M., Shafiq, M., ... \& Gillies, R. J. (2018). Data for NSCLC radiogenomics collection. The Cancer Imaging Archive.
4. Dong, Y., Feng, Q., Yang, W., Lu, Z., Deng, C., Zhang, L., ... \& Cui, C. (2016). Preoperative prediction of sentinel lymph node metastasis in breast cancer by radiomic signatures from dynamic contrast-enhanced MRI. *Journal of Magnetic Resonance Imaging*, 44(6), 1468-1477.
5. Hu, Y., Xie, C., Yang, H., Ho, J. W. K., Wen, J., Han, L. J., ... \& Chen, W. (2019). Assessment of intratumoral heterogeneity and prognosis using CT textural features in advanced lung adenocarcinoma treated with epidermal growth factor receptor tyrosine kinase inhibitors. *American Journal of Roentgenology*, 213(3), W176-W184.
6. 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.
7. Aerts HJWL, Velazquez ER, Leijenaar RTH, et al. Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. *Nat Commun*. 2014; 5:4006.
8. Yip, S. S. F., \& Aerts, H. J. W. L. (2016). Applications and limitations of radiomics. *Physics in Medicine \& Biology*, 61(13), R150-R166.
9. P. Afshar, A. Mohammadi, K. N. Plataniotis, A. Oikonomou and H. Benali, "From Handcrafted to Deep-Learning-Based Cancer Radiomics: Challenges and Opportunities," in *IEEE Signal Processing Magazine*, vol. 36, no. 4, pp. 132-160, July 2019, doi: 10.1109/MSP.2019.2900993. keywords: Cancer; Feature extraction; Tumors; Predictive models; Medical diagnostic imaging; Deep learning; Machine learning.
10. 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.
11. 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.
12. 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).

13. Chen, B., Yang, L., Zhang, R., Luo, W., \& Liu, W. (2020). Radiomics: An overview in lung cancer management—a narrative review. *Annals of Translational Medicine*, 8(11), 712.
14. Jia Weikuan, Sun Meili, Lian Jian, and Hou Sujuan, Feature dimensionality reduction: a review, *Complex & Intelligent Systems* 8 (2022), 2198–6053.
15. Michael Berry and Azlinah Mohamed, Supervised and unsupervised learning for data science, 01 2020.
16. Van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(Nov), 2579-2605.
17. Laurens van der Maaten, Geoffrey Hinton. Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research*, 2008.
18. Awadallah, M. (2018). Staging and treatment implications of the 7th vs. 8th AJCC staging system for oral cancer. *Journal of Oral and Maxillofacial Surgery*.
19. Edge, S.B., Compton, C.C. The American Joint Committee on Cancer: the 7th Edition of the AJCC Cancer Staging Manual and the Future of TNM. *Ann Surg Oncol* 17, 1471–1474 (2010).
20. 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.
21. Lazebnik, Y. (2010). What are the hallmarks of cancer? *Nature Reviews Cancer*, 10(2), 232-239. <https://doi.org/10.1038/nrc2839>.
22. Souza, C., Fonseca, L., Mendonça, V., & Gomes, E. (2012). Lifestyle factors and cancer risk: The role of genetics. *Cancer Epidemiology, Biomarkers & Prevention*, 21(7), 1235-1241. <https://doi.org/10.1158/1055-9965.EPI-12-0042>.
23. Finn, O. J. (2012). Cancer immunology. *New England Journal of Medicine*, 366(15), 1481-1490. <https://doi.org/10.1056/NEJMra1202522>.
24. Magalhães, L. G., Rodrigues, V., & da Silva, C. G. (2018). Personalized medicine and cancer therapy. *Oncology Letters*, 15(4), 5151-5159. <https://doi.org/10.3892/ol.2018.8124>.
25. C Rencher Alvin, Methods of multivariate analysis, John Wiley&sons inc, Publication, Canada (2002).

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Please feel free to contact the authors at:

- Victor Tolulope Aderanti: [aderanti@etsu.edu](mailto:aderanti@etsu.edu)
- Mostafa Zahed: [zahedm@etsu.edu](mailto:zahedm@etsu.edu)

## APPENDIX

### Abbreviations

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

PET Positron Emission Tomography

CT Computed Tomography

MRI Magnetic Resonance Imaging

MSE Mean Square Error

AIC Akaike Information Criterion

BIC Bayesian Information Criterion

DICOM Digital Imaging and Communications in Medicine

XML Extensible Mark-up Language

OS Overall Survival

PFS Progression-Free Survival

NSCLS Non-Small Cell Lung Cancer

PD-L1 Programmed Death L1igand 1

GLCM Gray Level Co-occurrence Matrix

GLRLM Gray Level Run Length Matrix

GLZLM Gray Level Zone Length Matrix

NGTDM Neighborhood Gray Tone Difference Matrix

NMF Non-negative Matrix Factorization

FSA Feature Selection Algorithm

FEA Feature Extraction Algorithm

### SAS CODE

```
FILENAME REFFILE '/home/u63178976/Thesis/Lung_Cancer_PET_norm.xls';
```

```
PROC IMPORT DATAFILE=REFFILE
```

```
DBMS=XLS
```

```
OUT=Lung;
```

```
GETNAMES=YES;
```

```
RUN;
```

```
proc print data=Lung;
```

```
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;
```

```
run;
```

```
/*view mean and standard deviation of normalized dataset*/
```

```
proc means data=normalized_Lung Mean StdDev ndec=2;
```

```
*var values;
```

```
run;
```

```
/* Perform PCA using PROC PRINCOMP */
```

```
proc princomp data=normalized_Lung plot;
```

```
run;
```

```
proc factor data=normalized_Lung simple corr score scree;  
run;
```

```
proc princomp out=normalized_Lung plots= score(ellipse ncomp=6); run;
```