Joanna Matuszak 255762, Joanna Wojciechowicz 255747

Microarray brain tumor diagnostics

February 9, 2024

Contents

1. Introduction

In this phase of our research, we continue with further exploration of the brain tumor microarray dataset, which comprises data related to five distinct types of brain tumors. Notably, this dataset is characterized by its large number of features in oposition to a very limited number of observations. In this part, we investigate the application of selected dimensionality reduction techniques — PCA (Principal Component Analysis) and MDS (Multidimensional Scaling). We aim to visualize the outcomes of these methods. Subsequently, we integrate them into our classification assessment and contrast the results with the previous phase where feature selection was involved instead of dimensionality reduction.

In our classification analysis, we consider a range of algorithms, including K Nearest Neighbors, Linear and Quadratic Discriminant Analysis, Multinomial Logistic Regression, Decision Tree, and Random Forest. For each algorithm, we perform parameter tuning utilizing leave-one-out cross-validation. We then compare our models, optimized with chosen parameters, based on misclassification errors and macro-averaged F1 score.

Moving forward, we transition to cluster analysis with a focus on quality assessment. Here, we compare the outcomes of approaches involving dimensionality reduction and unsupervised feature selection using the medoids from PAM (Partitioning Around Medoids) algorithm. We evaluate the performance of selected partitioning and hierarchical methods, namely KMeans, PAM, DIANA, and AGNES.

Additionally, we offer suggestions for further research directions that could enhance our findings. Our primary research objectives revolve around comparing feature selection with dimensionality reduction in the context of classification results. We also aim to uncover underlying patterns within our dataset through cluster analysis and compare the effectiveness of clustering with dimensionality reduction and unsupervised feature selection methods. Furthermore, we explore whether the results of our cluster analysis align with predefined classes. Our research potentially leads to improved diagnostic methods that provide tumor type suggestions based on various classification and clustering approaches. This study holds promise for aiding medical professionals in making more informed decisions regarding tumor diagnosis and treatment.

2. Dimensionality reduction

In addressing the challenge of large dimensionality within our dataset, we opt for a dimensionality reduction approach rather than feature selection. We apply both PCA (Principal Component Analysis) and MDS (Multidimensional Scaling) methods for this purpose. Notably, since our dataset exclusively comprises quantitative features and we utilize Euclidean distance in the MDS method, it essentially aligns with PCA (with precision in terms of sign). Nonetheless, we analyze both techniques. Subsequently, we undertake classification and clustering tasks with use of the principal components derived from PCA on the learning set and the whole data respectively.

2.1. Principal Component Analysis (PCA)

In PCA, we extract the initial principal components that collectively explain 95% of the total data variability. Initially, we partition the data into learning and test sets, conducting dimensionality reduction on the learning one and apply the results on the test one. From our learning set, we extract 26 principal components, marking a significant reduction from the over 5,000 features initially present.

A comparison of the variance explained by each principal component is illustrated in Figure 2.1, where only 30 components are visible, as further components lack significant variance explanatory capability. Notably, to achieve a 95% explanation of variability, we select the initial 26 components (2.2). Additionally, it is noteworthy that differences in variance explainability are most pronounced in the first few components.

Figure 2.3 showcases the features contributing most significantly to the first principal component, with feature V3260 occupying the foremost position. Similarly, for the second principal component, feature V588 emerges as pivotal (2.4).

2.2. Multidimensional scaling (MDS)

Subsequently, MDS is performed on our learning set, aiming to preserve the original distances between objects. Figure 2.5 compares the representations obtained from MDS and PCA, revealing agreement between the two methods, with precision in terms of sign. Evaluation of MDS performance and its accuracy in preserving original distances is conducted through Figure 2.6, which presents Shepard diagrams and normalized STRESS values for various dimensions. Across the 30 rows of our learning data, the reduction in STRESS value is observed to be consistent (what we can observe on the

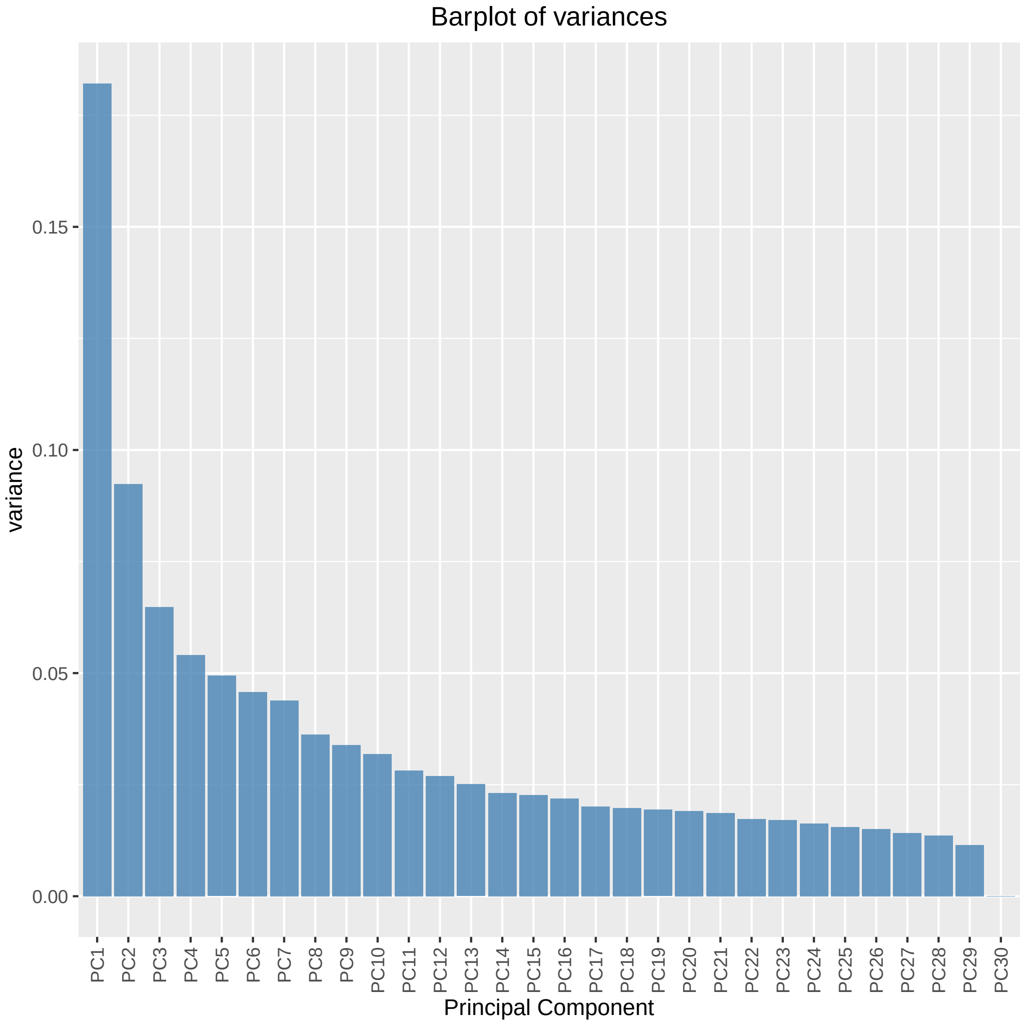

Figure 2.1. Amount of variance explained by subsequent principal components

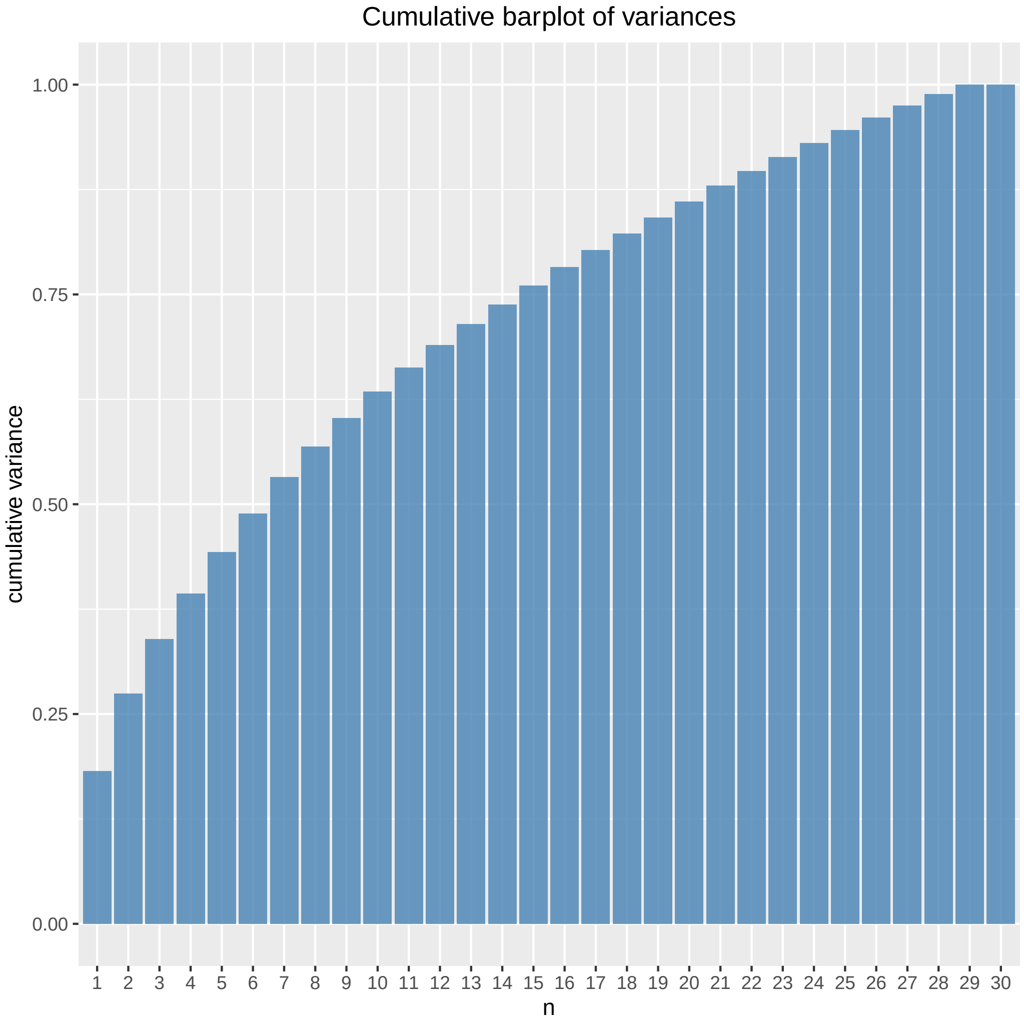

Figure 2.2. Cumulative amount of variance explained by first n components

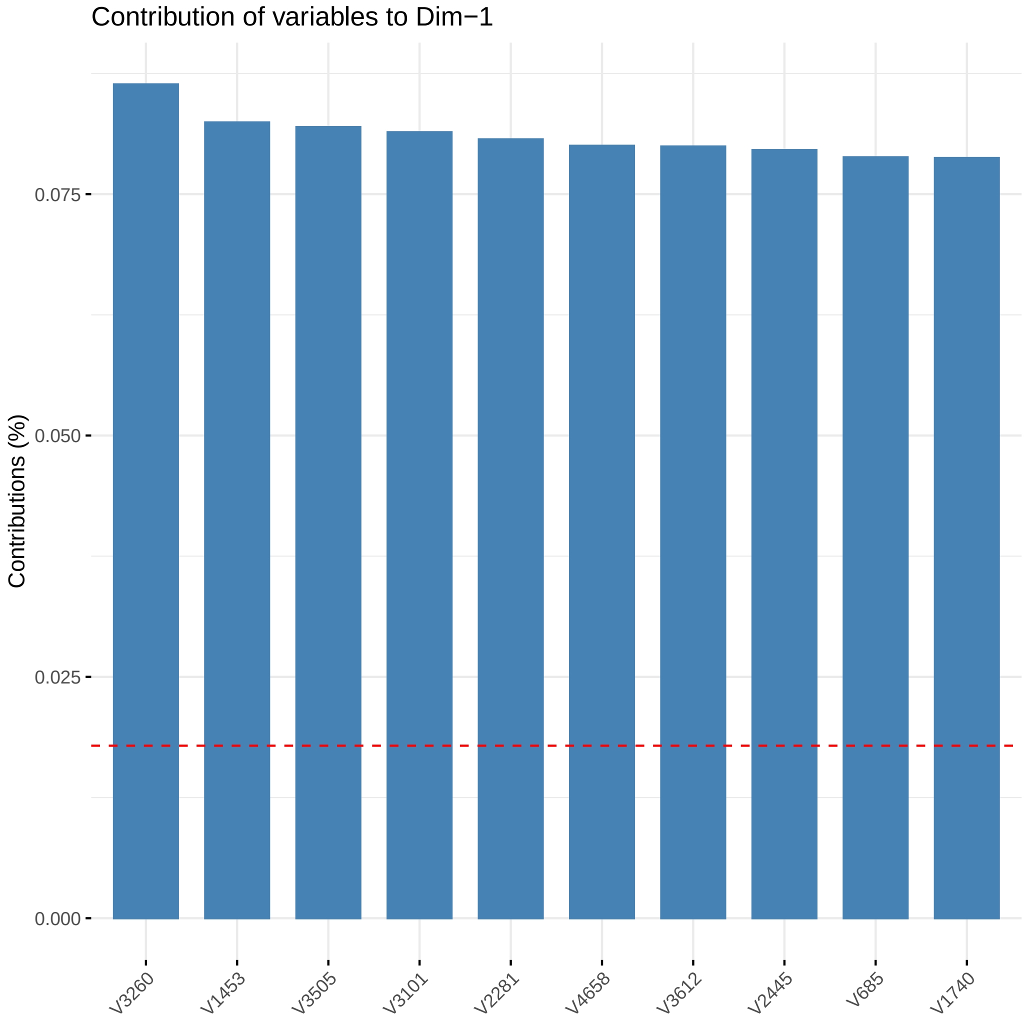

Figure 2.3. Contribution of 10 most important variables to PC1

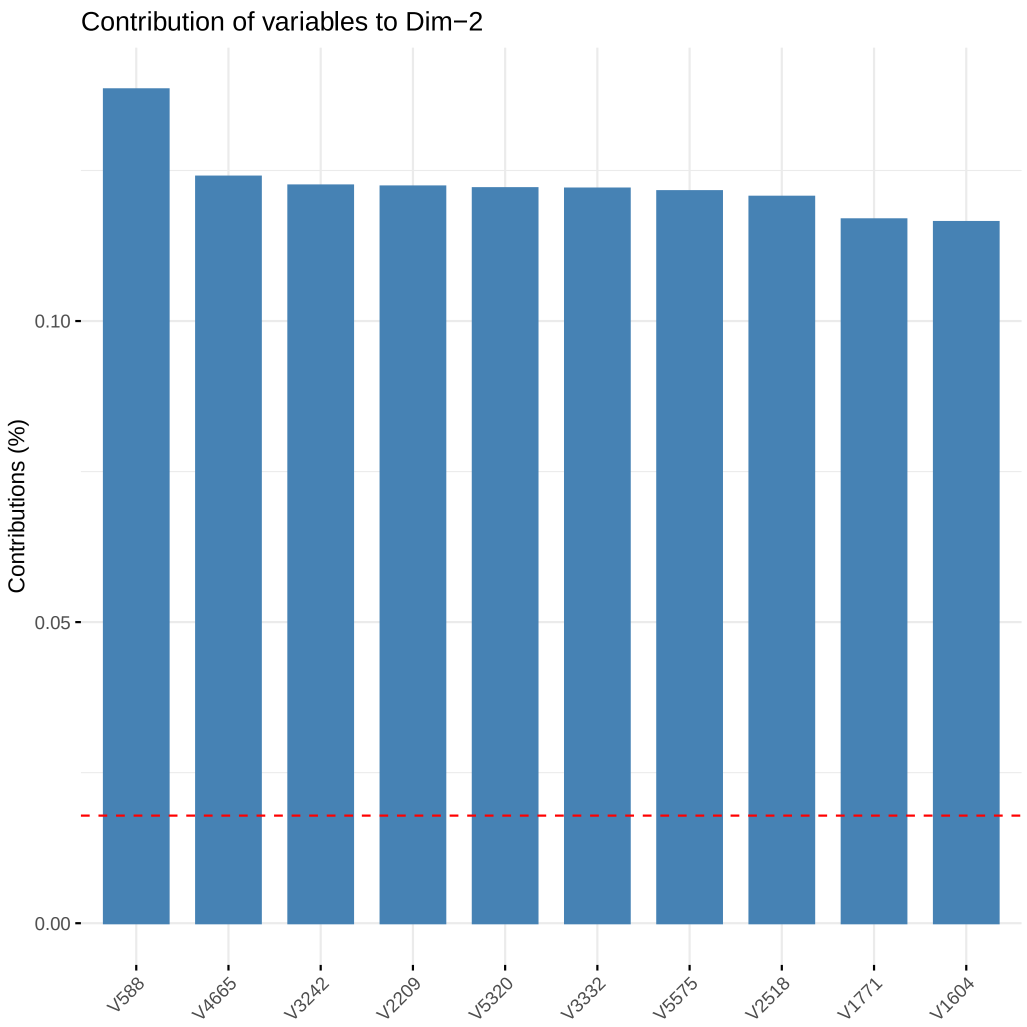

Figure 2.4. Contribution of 10 most important variables to PC2

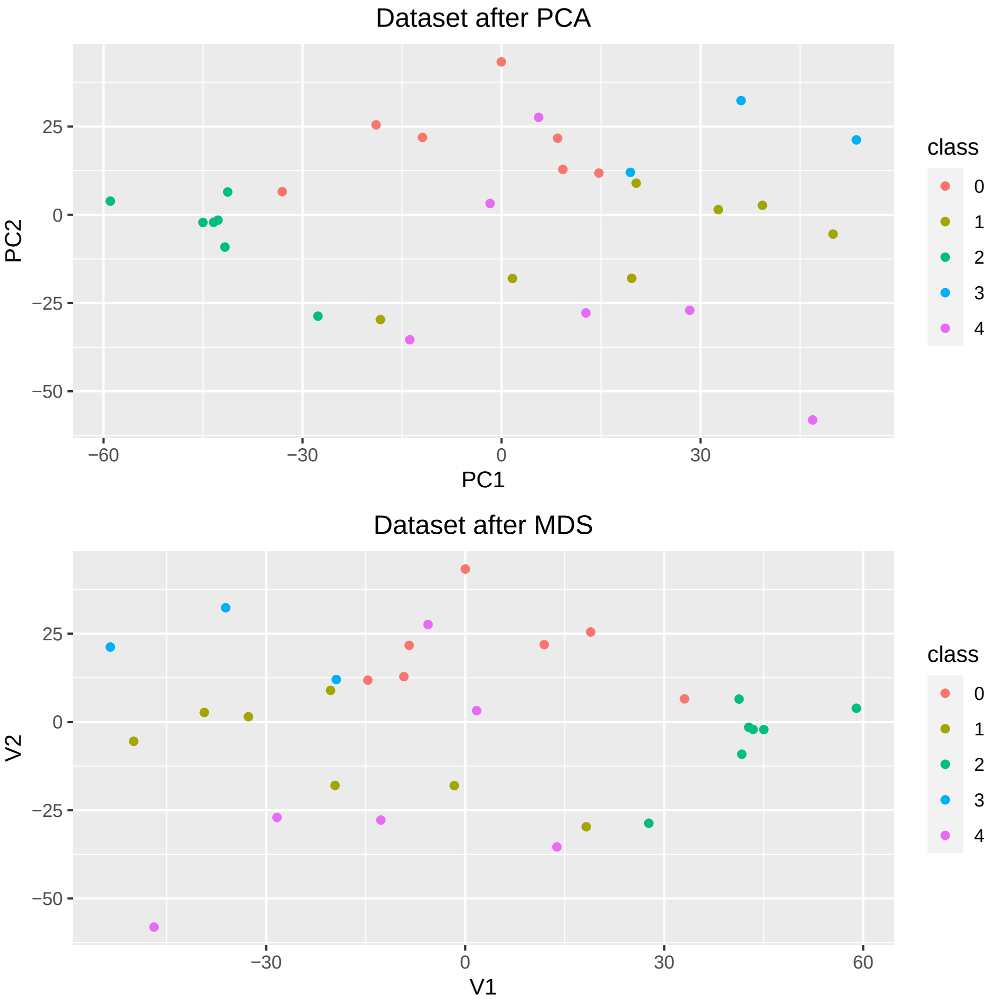

Figure 2.5. Comparison of new coordinates after PCA and LDA

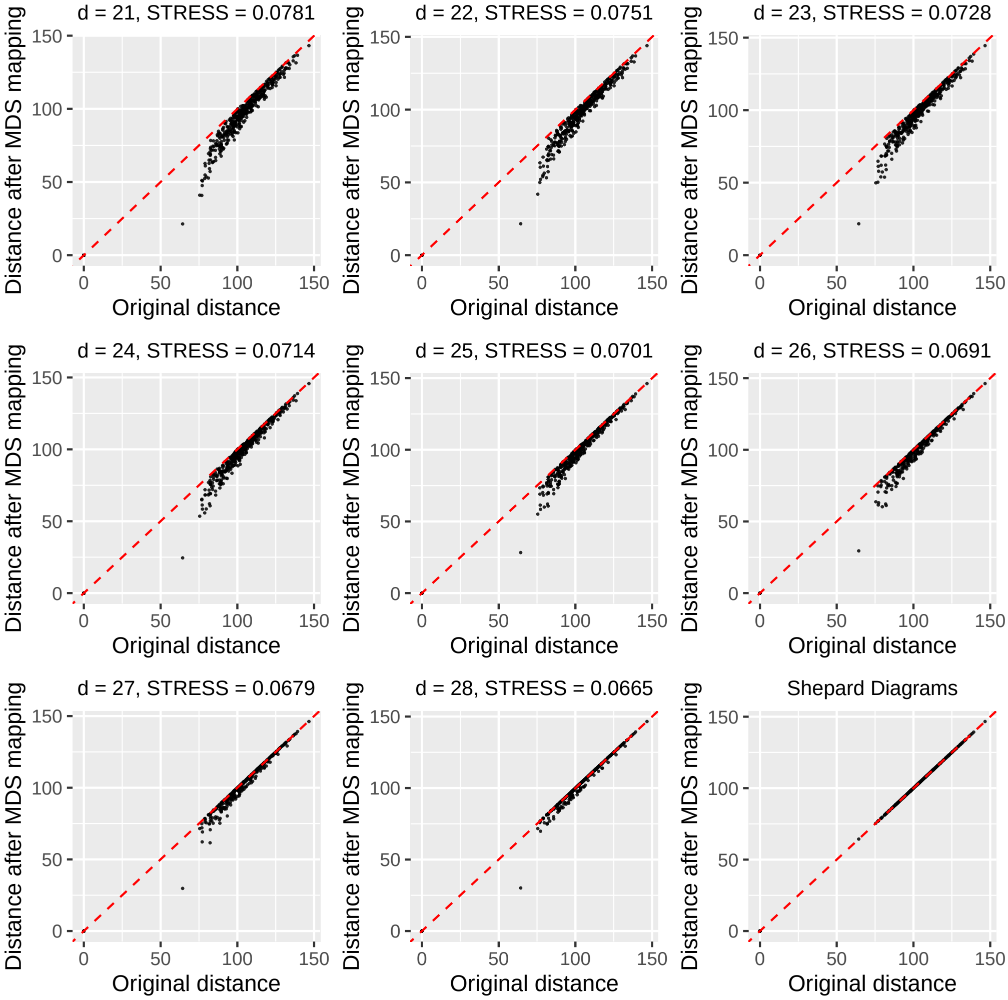

Figure 2.6. Shepard diagram

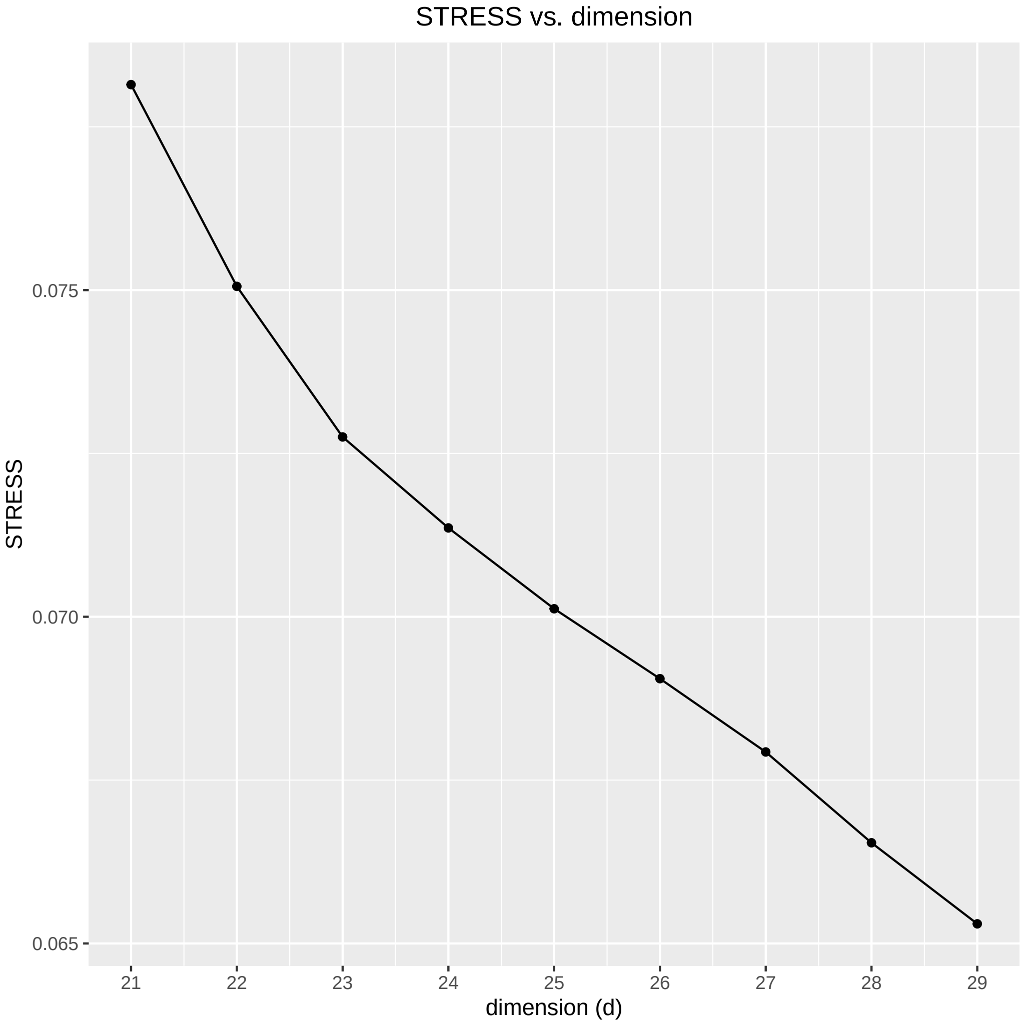

Figure 2.7. STRESS value for given dimension

Figure 2.7), indicative of improved preservation of original distances as we approach this number.

3. Clasification with dimensionality reduction

Once again, we face a problem of multiclass classification. Our investigation encompasses the following models:
— K Nearest Neighbors,
— Linear and Quadratic,
— Discriminant Analysis,
— Multinomial Logistic Regression,
— Decision Tree,
— Random Forest.

Moreover, we will compare our prior results employing feature preselection with our current approach involving dimensionality reduction.

To tackle the classification task, we initially partition the data into learning (2/3) and test (1/3) sets, considering the class imbalance issue and employing stratified sampling. For parameter tuning of the chosen models, we employ leave-one-out cross-validation. Additionally, we apply data standardization before executing the PCA procedure. Following this, we refrain from standardizing the principal components to preserve the results obtained from PCA, with the expectation of achieving a meaningful separation of classes.

Subsequently, upon selecting the optimal models, we proceed to compare their misclassification errors and macro-averaged F1 scores.

3.1. K Nearest Neighbors

We will begin our analysis with the K Nearest Neighbors (KNN) algorithm. To determine the optimal parameter k, we will compare the average misclassification errors using leave-one-out cross-validation across a range of k values, up to 10 nearest neighbors.

From the figure 3.1, it is evident that as we increase the number of neighbors considered, the error generally increases. However, this trend exhibits some fluctuations. After careful examination, we identify k=1 as the optimal parameter value. This choice is based on achieving the smallest misclassification error on the validation set. For k = 1 we get the smallest error on the train set but since we also have the smallest error on the validation set we will not consider this case as an overfitting.

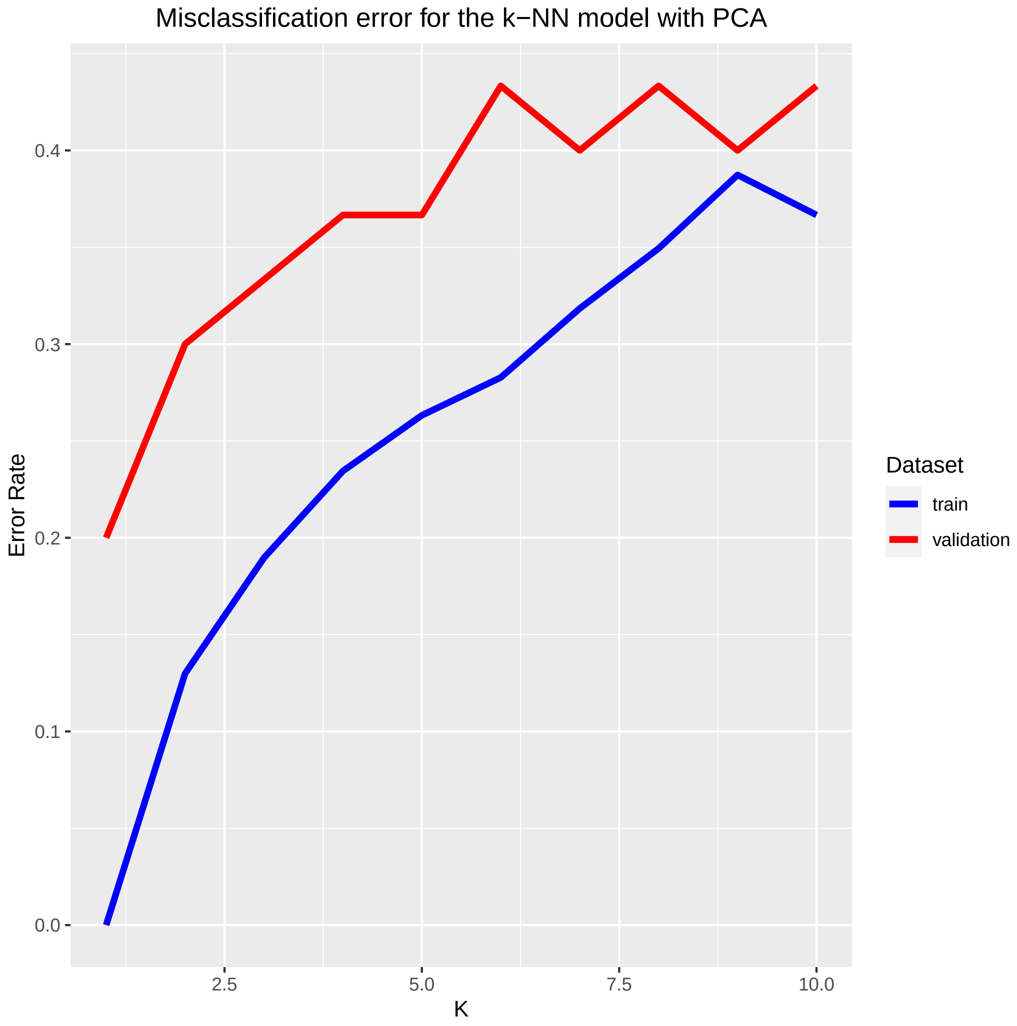

Figure 3.1. Misclassification errors for train set and validation set for the kNN model with PCA

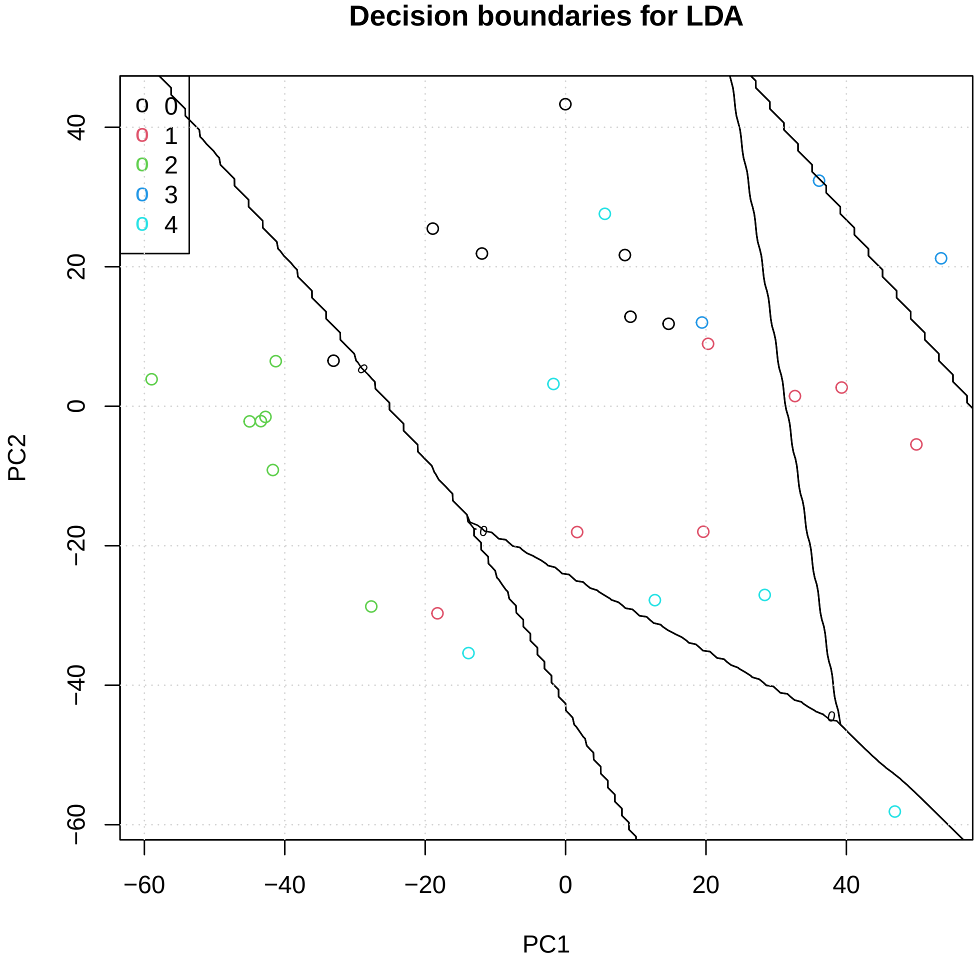

Figure 3.2. Decision boundaries for LDA

3.2. Linear Discriminant Analysis and Quadratic Discriminant Analysis

Moving forward, let's delve into two linear classification techniques: Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). Both methods operate under the assumption of multivariate normality within the dataset.

With LDA, we assume uniform covariance matrices across all classes, whereas with QDA, we estimate individual covariance matrices for each class. Given our current approach utilizing PCA, where we are working with only 26 principal components and 30 samples in the learning set, LDA is suitable for our analysis. However, for QDA, the assumption of distinct covariance matrices for each class renders our sample size insufficient for meaningful results, as it requires a larger dataset for accurate estimation.

3.3. Multinomial Logistic Regression

Another linear model that we will discuss is Multinomial Logistic Regression, which imposes fewer assumptions on data. In case of this model we will not search for optimal parameters.

3.4. Decision Tree

Next, we explore another classification algorithm: Decision Tree. For our dataset, we initialize the Decision Tree with parameters minsplit = 1 and minbucket = 1, which are reasonable for our data.

A crucial parameter to tune in Decision Trees is the complexity parameter (cp), which determines the size of the tree. We investigate cp values ranging from 0.02 to 0.2. Upon plotting the misclassification errors in Figure 3.3, we observe that validation errors are minimized for cp values below 0.08. Remarkably, we discern no signs of overfitting; instances with the lowest train errors correspond to the lowest validation errors.

For our final model, we opt for a complexity parameter of 0.08. A larger cp value results in a smaller tree, mitigating the risk of overfitting at the expense of increased bias. Thus, within the range of 0 to 0.08, we select the maximum value to strike a balance between model simplicity and generalization capability.

3.5. Random Forest

Now, we investigate the Random Forest algorithm, the most complex among the classification methods considered. We focus on examining the influence of the ntree parameter, which denotes the number of trees in the forest.

Initially, we cast a wide range to capture the appropriate range, spanning from 2 to 2^{14}. Upon analyzing the results depicted in Figure 3.4, we discern that at ntree = 2^{10}, the misclassification error for the validation set is minimized.

Further refinement of the ntree parameter within the vicinity of this value (Figure 3.5) reveals that ntree= 1000 yields the lowest misclassification error on the validation set, while also achieving a training error of 0.

3.6. Comparison of feature selection approach and dimensionality reduction approach

In the previous sections, we meticulously selected models and their parameters. Now, we proceed to train them on the previously generated learning set and evaluate their performance on the test set. The results of the LDA algorithm are presented in the graph (3.2). Each model's performance is compared between the current approach utilizing PCA and the previous approach involving feature preselection.

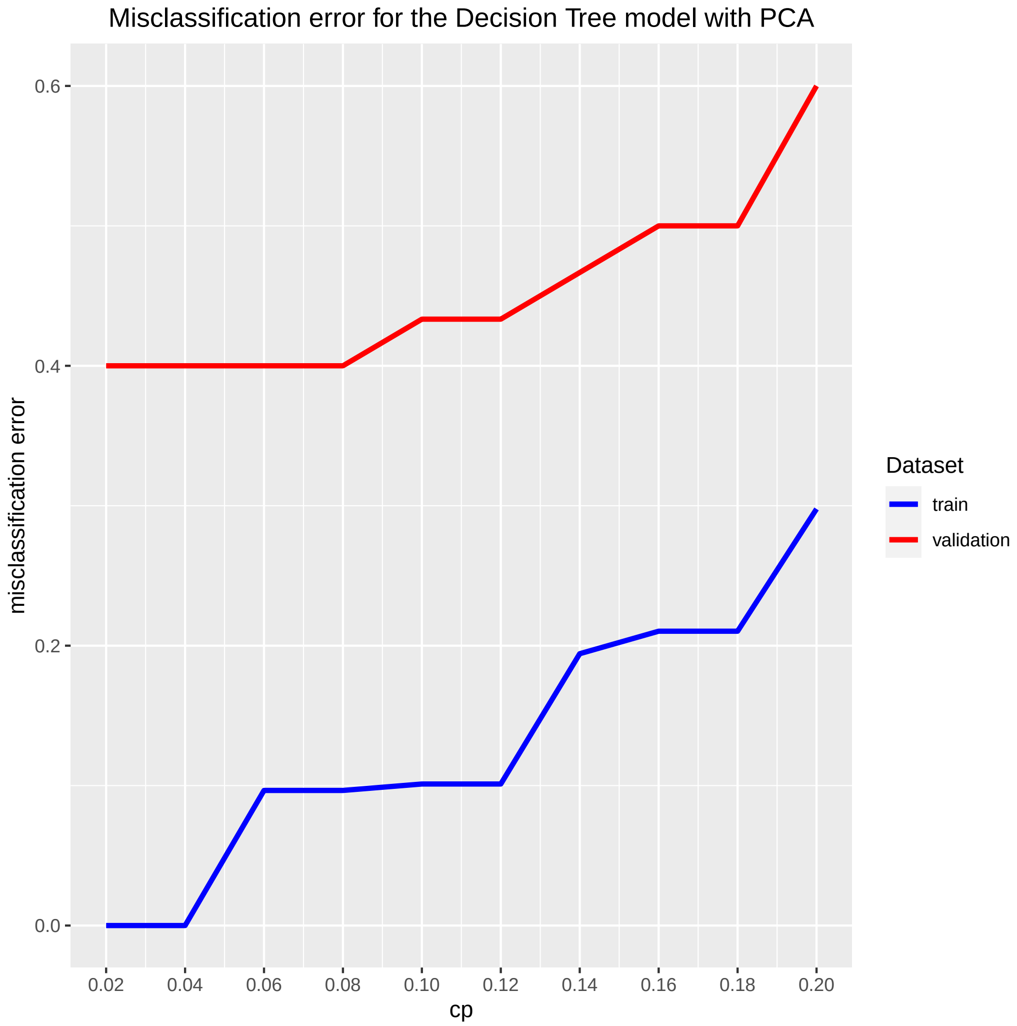

Figure 3.3. Misclassification errors for train set and validation set for the Decision Tree model with PCA

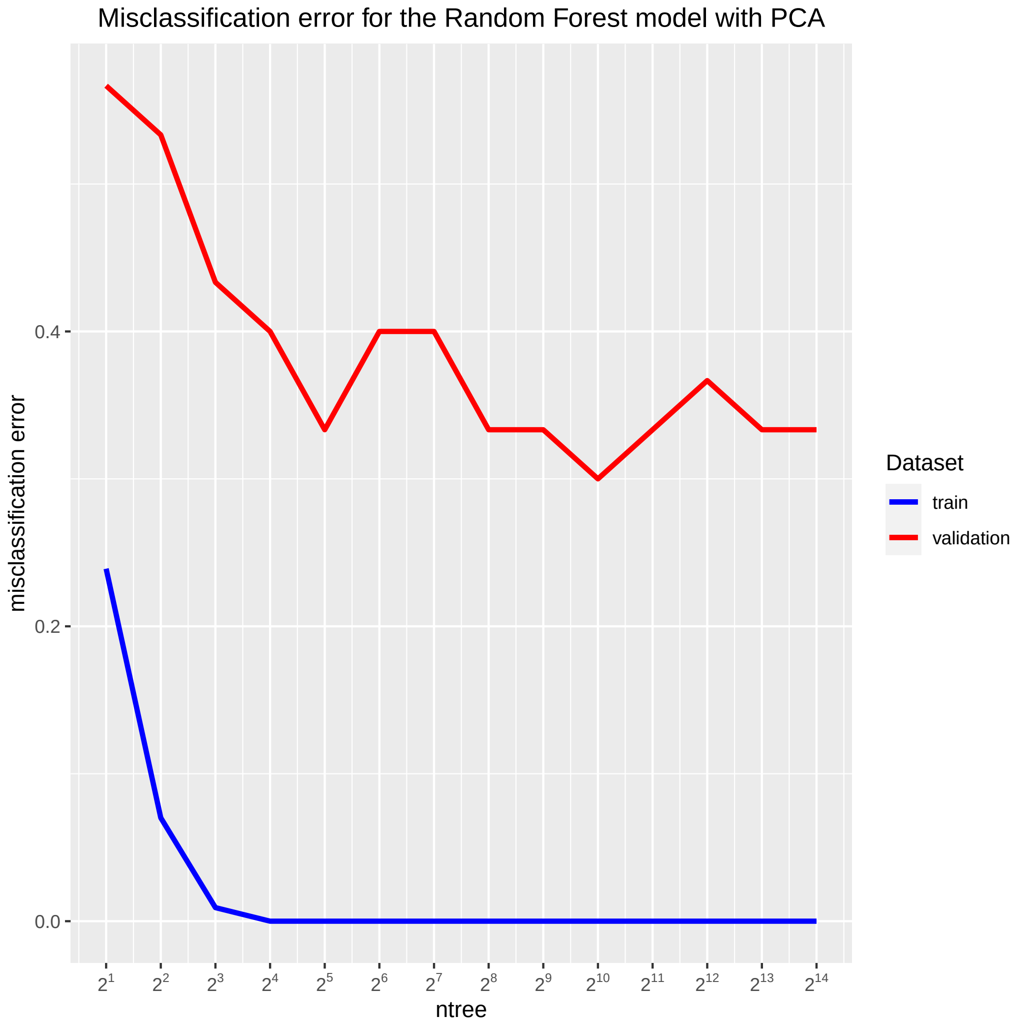

Figure 3.4. Misclassification errors for train set and validation set for the Random Forest model with PCA

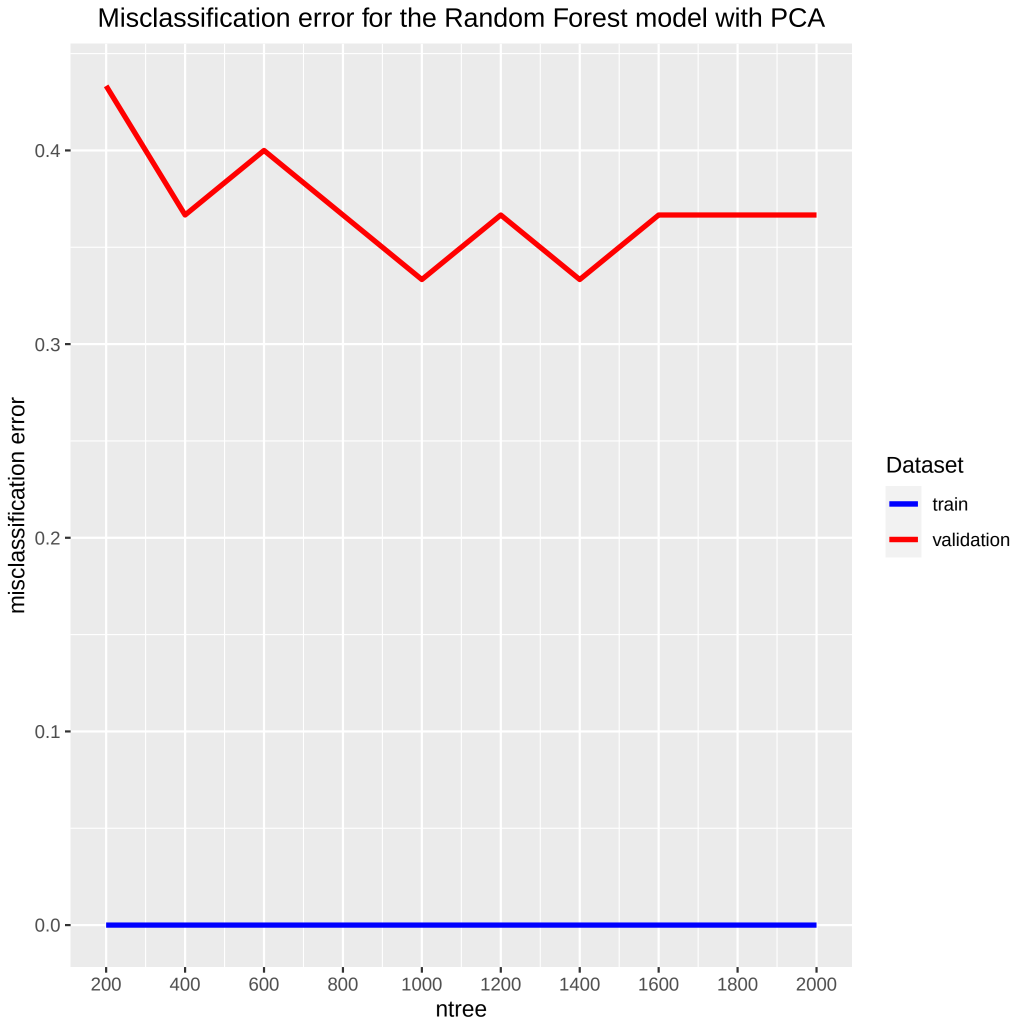

Figure 3.5. Misclassification errors for train set and validation set for the Random Forest model with PCA

model	error (PCA)
K Nearest Neighbors	0
Linear Discriminant Analysis	0
Multinomial Logistic Regression	0
Decision Tree	0.1
Random Forest	0

Table 3.1. Misclassification errors on the learning set for different classification models with PCA

Examining Table 3.1, we observe that, in terms of fitting the learning set, all models, except for Decision Tree, achieve zero learning error. The Decision Tree model exhibits a learning error of 0.1. However, our primary concern lies in the models' generalization ability, thus, we scrutinize their performance on the test set. Comparing the misclassification errors on the test set (Table 3.3), we find that all models perform well on unseen data, with Multinomial Logistic Regression leading with 0 test error.

Given the class imbalance issue, we address it by evaluating the macro-averaged F1 score (Table 3.5), which considers all classes equally. Here, Multinomial Logistic Regression emerges as the top performer, while LDA yields the lowest score. Notably, Multinomial Logistic Regression not only achieves the lowest error on the test set but also demonstrates superior handling of class imbalance.

Now, we compare the results across feature selection and dimensionality reduction. It's crucial to note that LDA was only performed in the PCA approach, and different parameters were tuned for Random Forest in these scenarios, making direct comparisons challenging.

Comparing Tables 3.1 and 3.2 for learning error, all models achieve zero, except for Decision Tree with PCA, which registers a 0.1 error. In Tables 3.3 and 3.4, we find that the PCA approach yields lower test error for kNN and Multinomial Logistic Regression. However, Decision Tree achieves the lowest error with the feature selection approach. Random Forest exhibits identical test errors for both PCA and feature selection approaches, but it's important to note the difference in tuned parameters.

Analyzing the changes in macro-averaged F1 scores (Tables 3.5 and 3.6), we notice higher F1 scores for kNN and Multinomial Logistic Regression with the PCA approach. Decision Tree with PCA failed to predict one class entirely and the F1 scores for this model with feature selection and with randomly chosen features were not high. Random Forest with PCA boasts a higher F1 score compared to the second approach.

model	misclassification error feature selection	misclassification error no feature selection
K Nearest Neighbors	0	0
Multinomial Logistic Regression	0	0
Decision Tree	0	0
Random Forest	0	0

Table 3.2. Misclassification errors on the learning set for different classification models with and without feature selection

model	misclassification error
K Nearest Neighbors	0.0833333
Linear Discriminant Analysis	0.25
Multinomial Logistic Regression	0
Decision Tree	0.3333333
Random Forest	0.1666667

Table 3.3. Misclassification errors on the test set for different classification models with PCA

model	misclassification error feature selection	misclassification error no feature selection
K Nearest Neighbors	0.1666667	0.1666667
Multinomial Logistic Regression	0.0833333	0.0833333
Decision Tree	0.25	0.5833333
Random Forest	0.1666667	0.4166667

Table 3.4. Misclassification errors on the test set for different classification models with and without feature selection.

model	F1 score
K Nearest Neighbors	0.9047619
Linear Discriminant Analysis	0.7866667
Multinomial Logistic Regression	1
Decision Tree	NA
Random Forest	0.8533333

Table 3.5. Macro-averaged F1 score on the test set for different models with PCA.

model	F1 score feature selection	F1 score no feature selection
K Nearest Neighbors	0.7933333	0.8333333
Multinomial Logistic Regression	0.9111111	NA
Decision Tree	0.6266667	0.4833333
Random Forest	0.8111111	0.6333333

Table 3.6. Macro-averaged F1 score on the test set for different models with and without feature selection

4. Clustering

In this phase of our project, we are conducting clustering analysis on our data. The clustering will be done with use of both partitioning and hierarchical methods, such as:

— K-means,
— Partition Around Medoids (PAM),
— Divisive Analysis (DIANA),
— Aglomerative Nesting (AGNES).

We will be exploring two different approaches. The first approach involves unsupervised feature selection utilizing the PAM algorithm. The second approach utilizes dimensionality reduction via the PCA method. We will perform the clustering as well as feature selection and dimensionality reduction on the whole dataset.

We will assess each clustering method using internal cluster validation indices, including:

— Total Within Sum of Square,
— Gap statistic,
— Average silhouette width,
— Connectivity,
— Dunn index.

For the first three indices, we will visualize the results for various numbers of clusters on graphs. Meanwhile, connectivity and the Dunn index will help us determine the optimal number of clusters. Additionally, in case of the hierarchical methods we will visualize the results via dendrograms as well.

Finally, we asses the agreement between the results of our cluster analysis and predefined classes.

4.1. Clustering with unsupervised feature selection

To execute unsupervised feature selection, we apply the PAM algorithm to the standardized features across all data points. We divide the data into 36 clusters and select the medoids of these clusters as our new feature set. The decision to have 36 clusters is not arbitrary. Through exploration, we discovered that in the case of dimensionality reduction applied to the entire dataset, 36 principal components are required to explain 95% of the variance. Consequently, we opted for the same number of features in both the unsupervised feature selection and dimensionality reduction approaches.

In Figure 4.1, we observe the internal validation measures computed for various numbers of clusters using the k-means algorithm. Notably, the total within sum of squares decreases as the number of clusters increases. Employing the elbow rule, we approximate the optimal number of clusters to

method	connectivity	clusters	Dunn	clusters
K Means	16.9333	3	0.4620	4
PAM	16.7663	2	0.4745	10
DIANA	14.2683	2	0.4450	10
AGNES	2.9290	2	0.5571	2

Table 4.1. Optimal scores of internal validation indicies for different clustering methods

be 2 based on this measure. However, the gap statistic suggests 1 cluster as optimal, while the average silhouette index achieves its highest value with 2 clusters.

Moving to the PAM algorithm (Figure 4.2), we find conflicting indications regarding the optimal number of clusters. The gap statistic and average silhouette width favors 2 clusters, whereas the elbow method suggests 3 clusters. Moreover, the average silhouette width shows no significant fluctuation for $k \in [7, 10]$.

In Figure 4.3, the results for the DIANA method are displayed. Once again, varying indices propose different optimal numbers of clusters, either 1 or 2.

For the AGNES algorithm, all three internal validation indices converge on 2 as the optimal number of clusters (4.4).

In most cases, the results were not consistent across different internal validation measures. We further evaluated the models using connectivity and Dunn index. From Table 4.1, we observe that the optimal number of clusters, considering connectivity, are 3, 2, 2, and 2, while for Dunn index, they are 4, 10, 10, and 2 for k-means, PAM, DIANA, and AGNES algorithms, respectively.

Taking into account all the results, we find that for all methods, the majority of indices suggest 2 as the optimal number of clusters.

4.2. Clustering with dimensionality reduction

Now, we proceed with the second approach, where we conducted PCA on the entire dataset for dimensionality reduction. We extracted 36 principal components to account for 95% of the variability.

Similarly, we evaluated these models using internal indices. In Figure 4.5, we observe that the optimal number of clusters for all cases is 2.

However, for the PAM results (Figure 4.6), each index suggests a different number of optimal clusters: 3 for total within sum of squares, 1 for gap statistic, and 9 for average silhouette width.

Similarly, for DIANA (Figure 4.7), the indices propose 2, 1, and 5 as optimal cluster numbers.

For AGNES (Figure 4.8), the indices suggest 2, 4, and 10 as optimal cluster numbers.

Table 4.2 displays the results for connectivity and Dunn index. In the case of connectivity, 2 is indicated as the optimal number of clusters, while

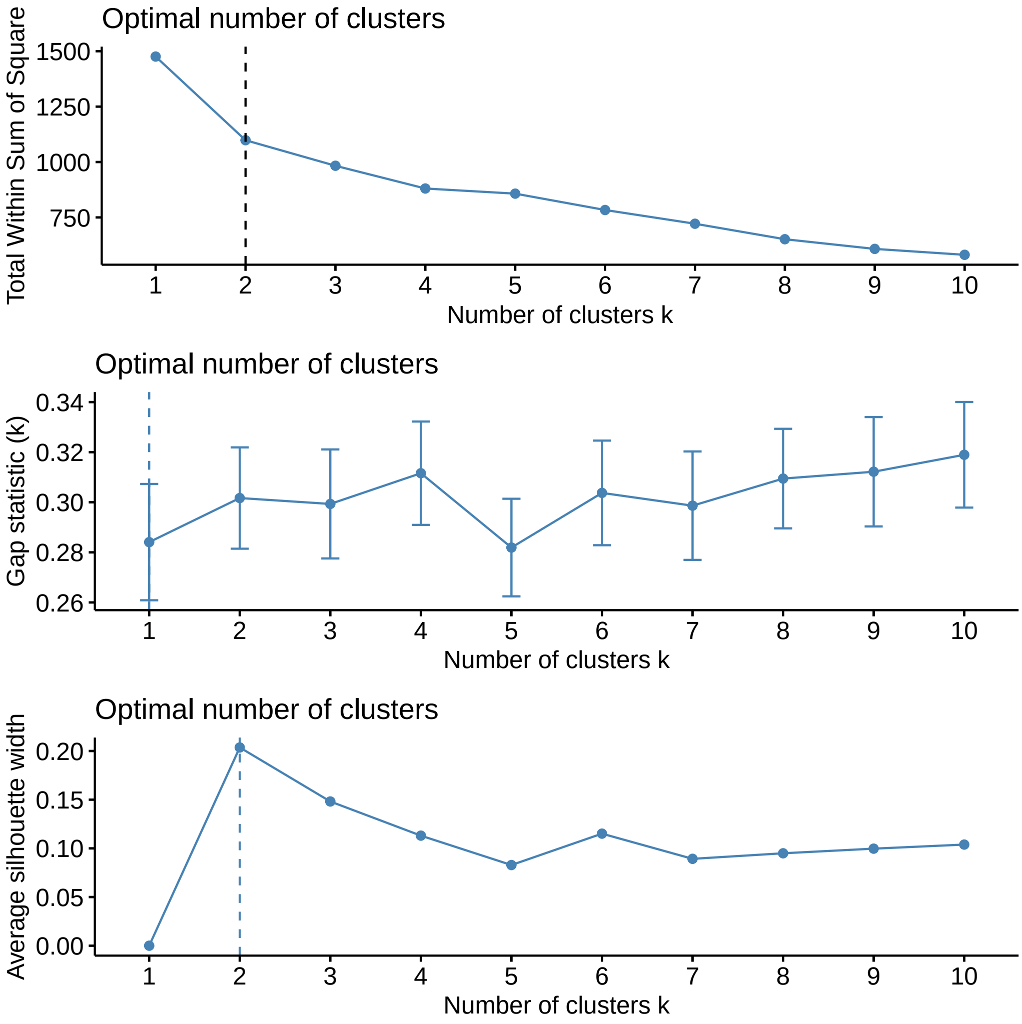

Figure 4.1. Internal validation measures for K Means clustering

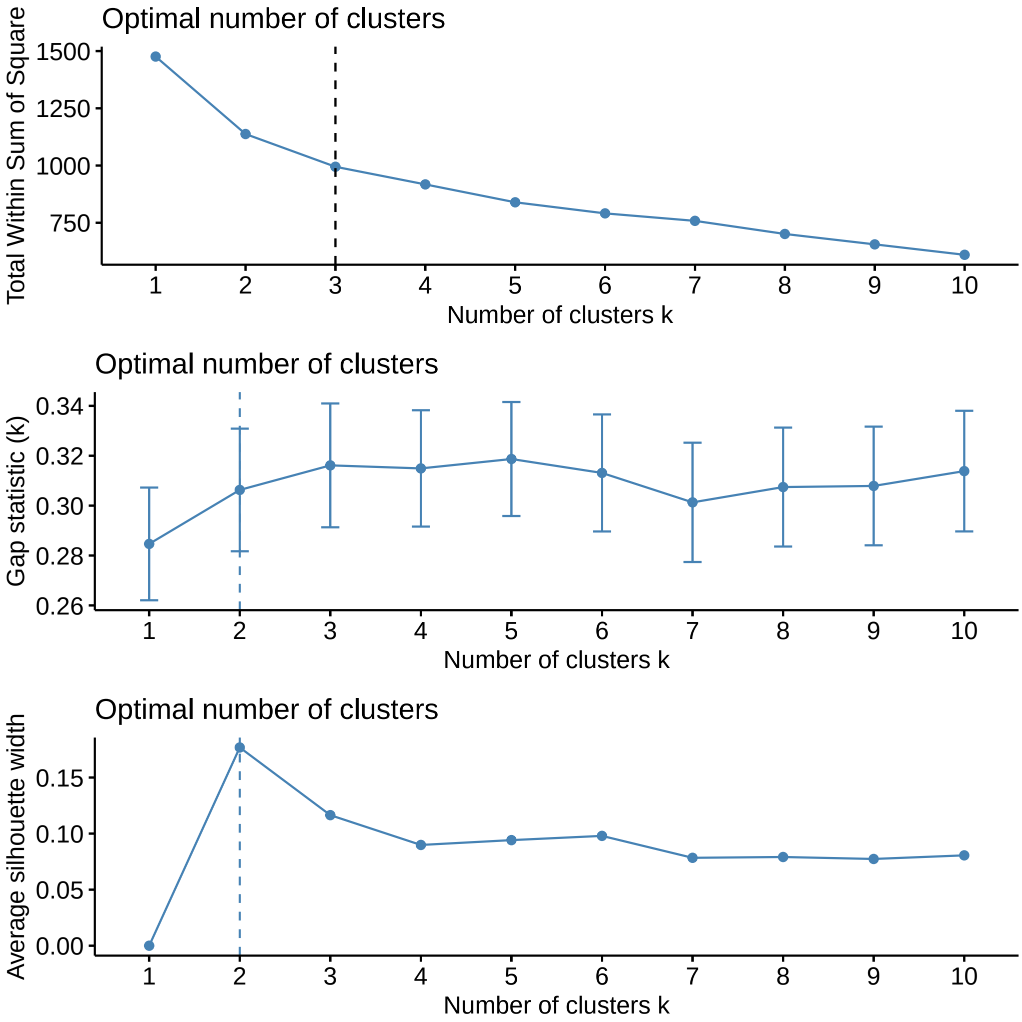

Figure 4.2. Internal validation measures for PAM clustering

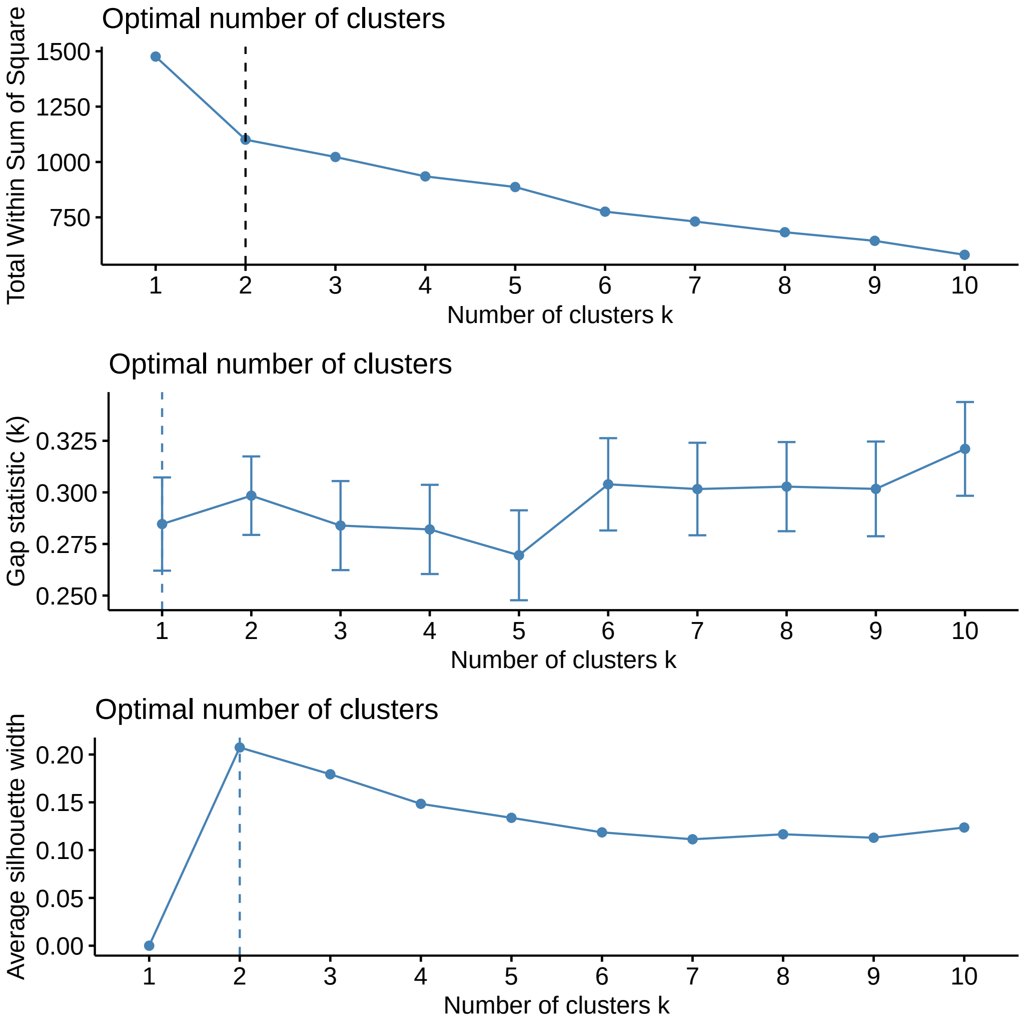

Figure 4.3. Internal validation measures for DIANA clustering

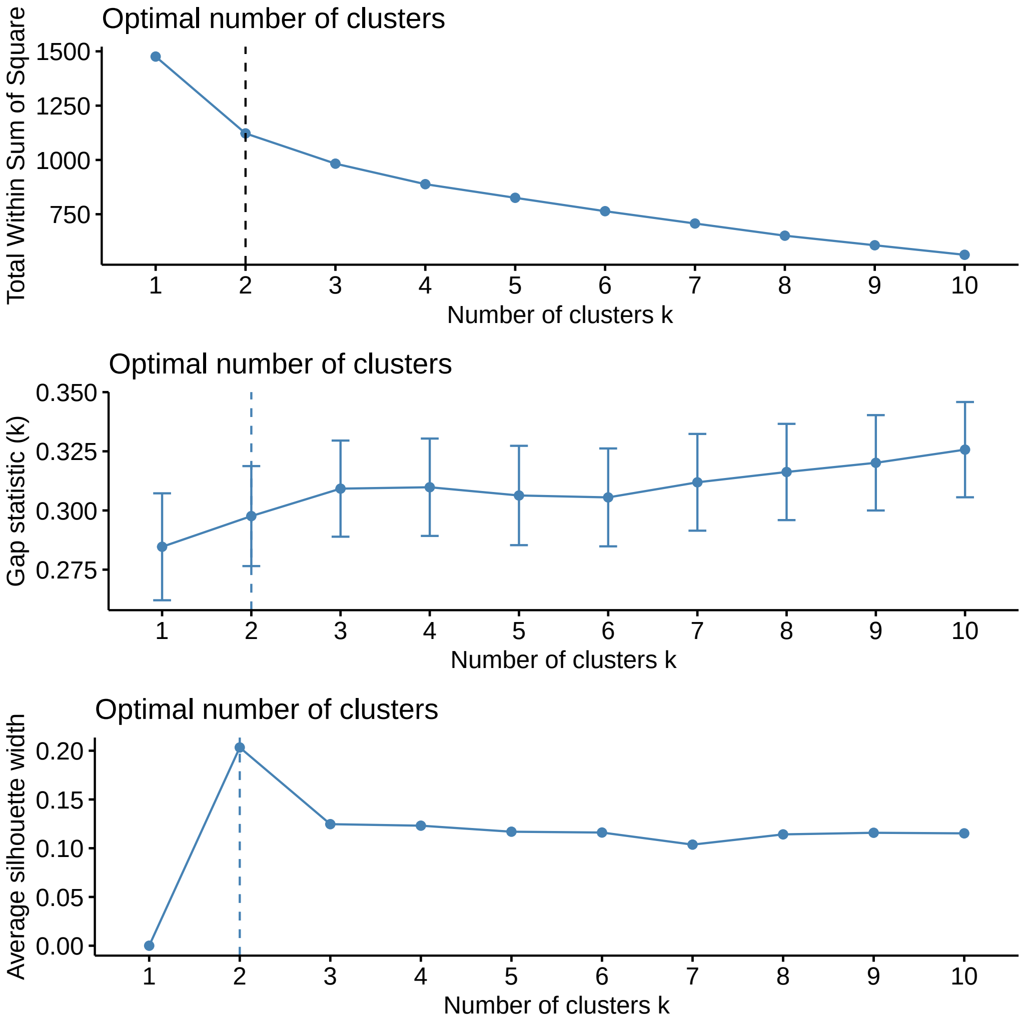

Figure 4.4. Internal validation measures for AGNES clustering

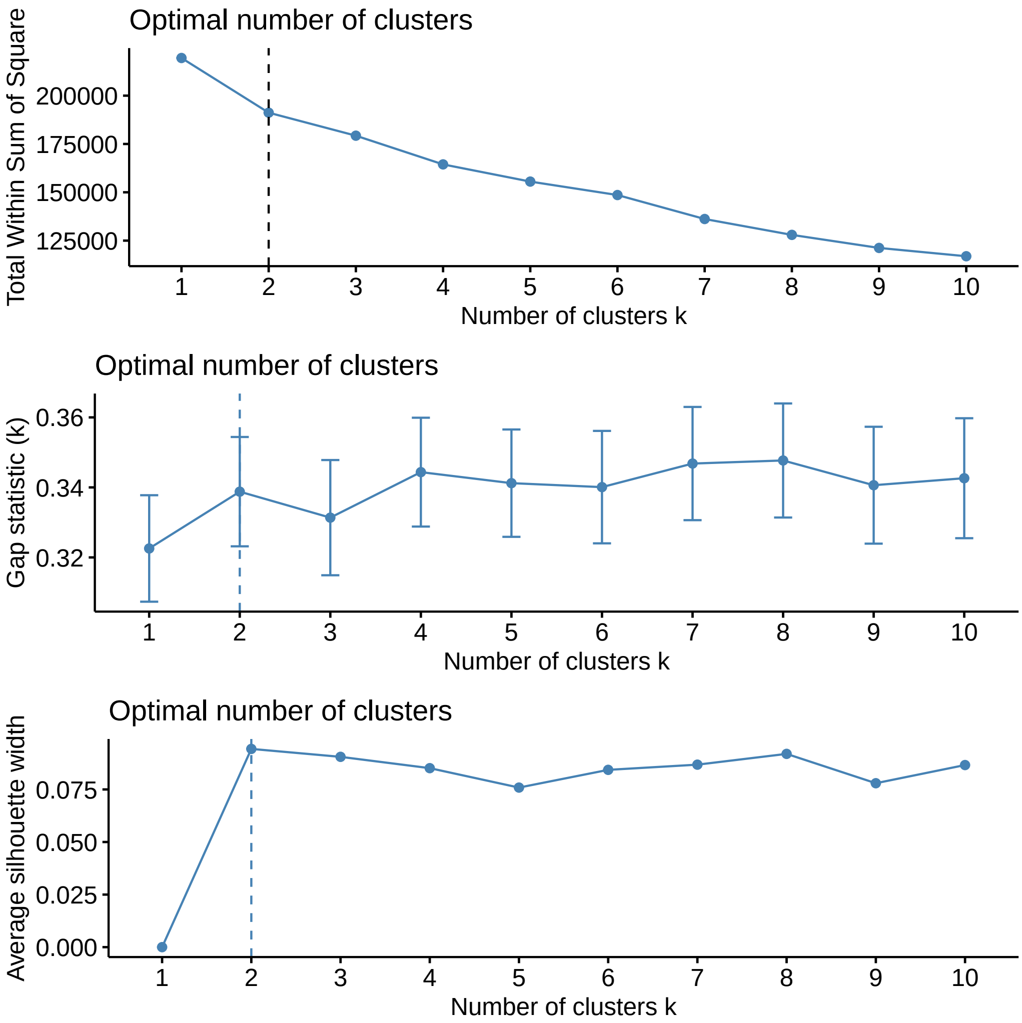

Figure 4.5. Internal validation measures for K Means clustering with dimensionality reduction

for the Dunn index, it is 10 for k-means and DIANA, 9 for PAM, and 6 for AGNES.

In summary, in clustering with dimensionality reduction, the majority of methods chose 2 as the optimal number of clusters for k-means, DIANA, and AGNES algorithms, and 9 for the PAM algorithm. However, it is important to note that the results were less consistent compared to clustering with unsupervised feature selection.

4.3. Comparison of unsupervised feature selection approach and dimensionality reduction approach

In this section, we aim to assess the agreement between the clustering results and the real classes of our data. To achieve this, we constructed models with a predefined number of clusters equal to the number of actual classes. We built these models using both unsupervised feature selection and dimensionality reduction approaches.

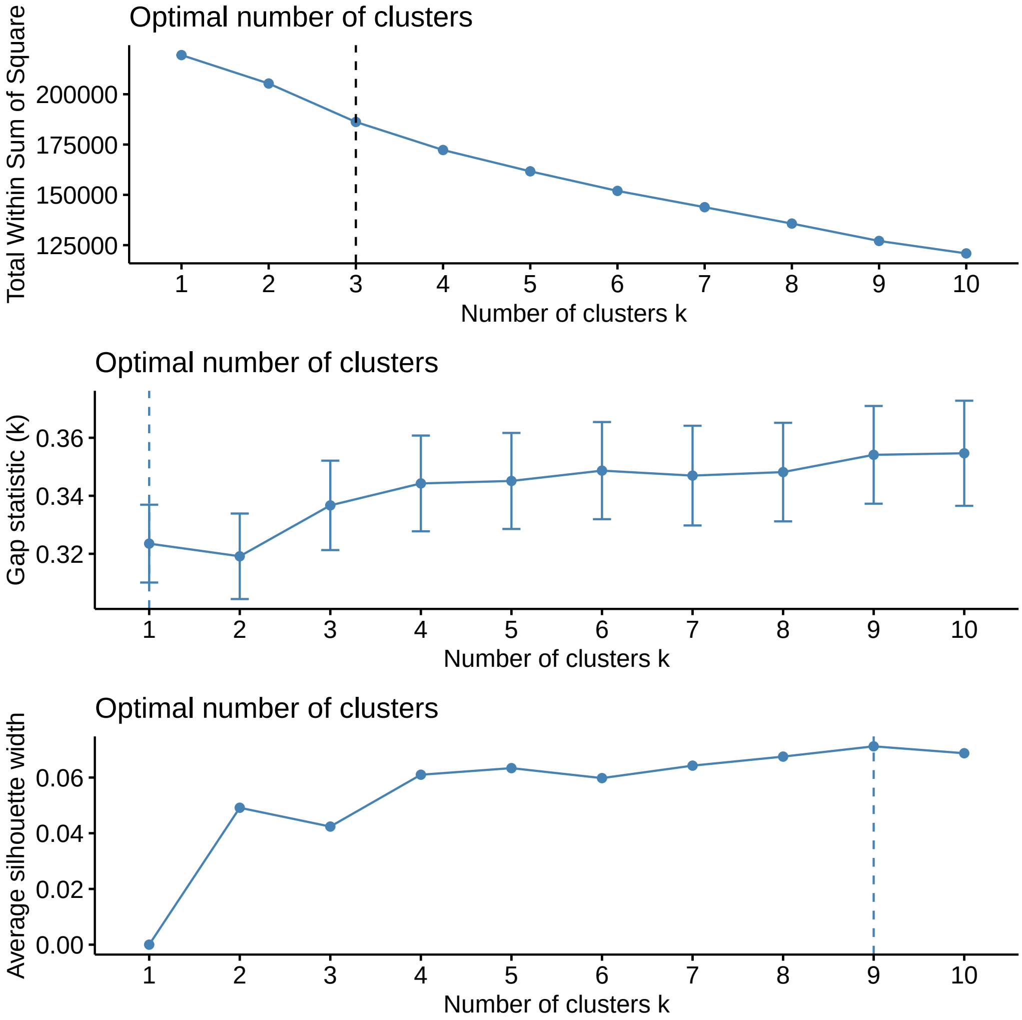

Figure 4.6. Internal validation measures for PAM clustering with dimensionality reduction

method	connectivity	clusters	Dunn	clusters
K Means	25.4464	2	0.6322	10
PAM	36.0079	2	0.5612	9
DIANA	18.9429	2	0.5860	10
AGNES	3.0956	2	0.7417	6

Table 4.2. Optimal scores of internal validation indicies for different clustering methods with dimensionality reduction

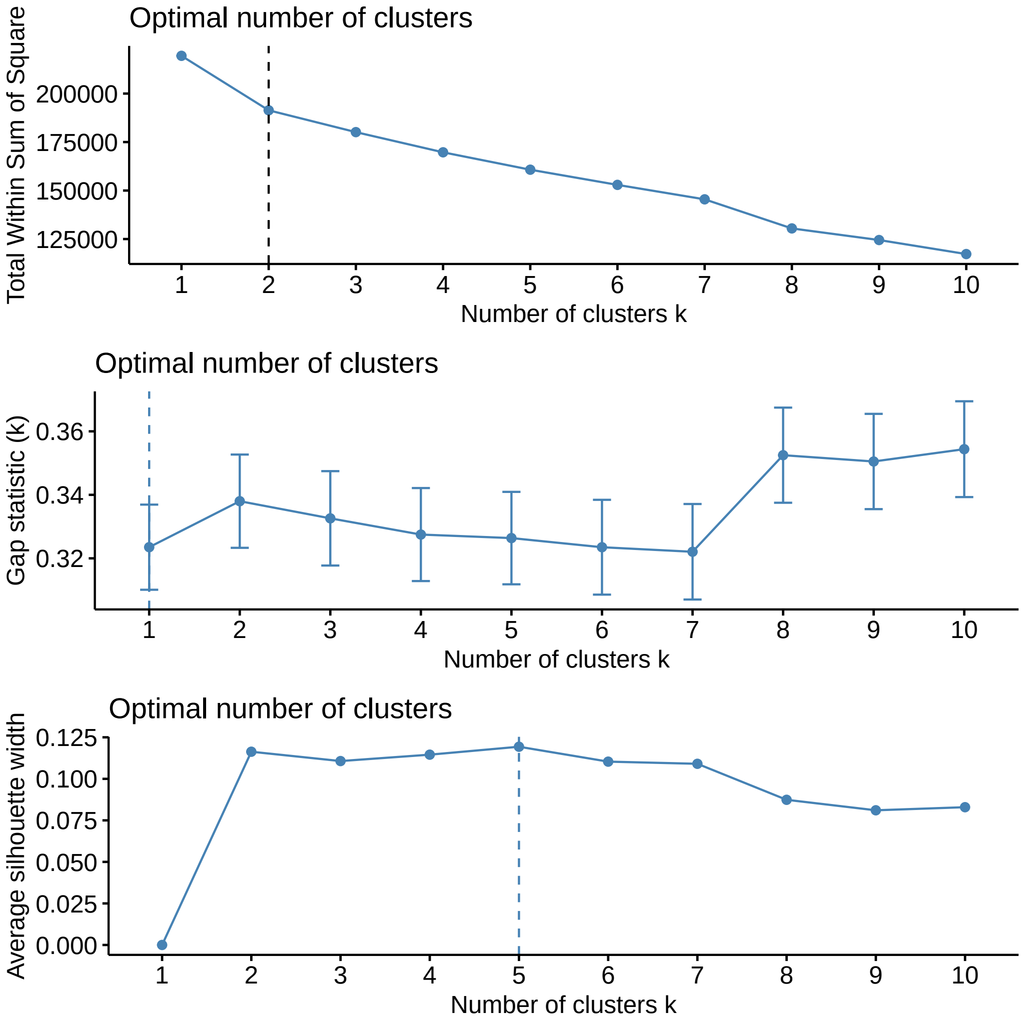

Figure 4.7. Internal validation measures for DIANA clustering with dimensionality reduction

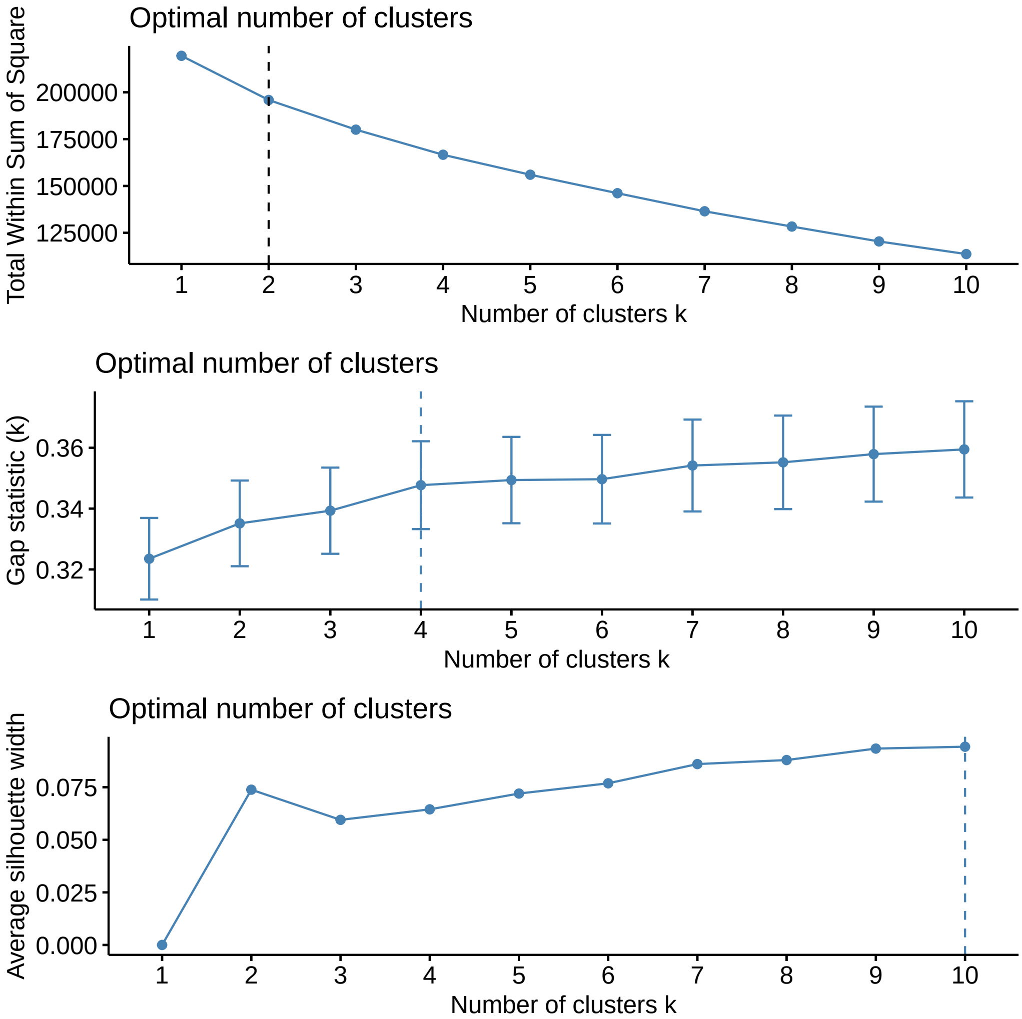

Figure 4.8. Internal validation measures for AGNES clustering with dimensionality reduction

method	unsupervised feature selection	dimensionality reduction
K Means	69.05%	64.29%
PAM	73.81%	66.67%
DIANA	50%	52.38%
AGNES	35.71%	30.95%

Table 4.3. Percentage of matches cases

In Figures 4.9 and 4.10, we present dendrograms for the DIANA and AGNES methods with unsupervised feature selection, while in Figures 4.11 and 4.12, we display dendrograms for the DIANA and AGNES methods with dimensionality reduction.

After constructing the models, we compare the results with the actual classes and calculate the percentage of matched cases (refer to Table 4.3). The findings reveal that the patterns detected by the PAM algorithm align quite well with the actual classes of our data, with matched cases percentages of 73.81% for unsupervised feature selection and 66.67% for dimensionality reduction approach. Conversely, the results of other algorithms seem to detect patterns that do not align with the actual classes. For instance, the AGNES results align with actual classes in only about 30% of cases. In general, the unsupervised feature selection approach resulted in higher percentage of matched cases that the dimensionality reduction one. The only exception to this rule are results for the DIANA algorithm.

Upon comparing the plots of average silhouette width, we observe that the unsupervised feature selection approach yielded higher values for all compared methods. Likewise, the total within sum of square values are lower for this approach. When examining the results of connectivity and Dunn index, similar conclusions arise - the unsupervised feature selection approach leads to lower values across the board.

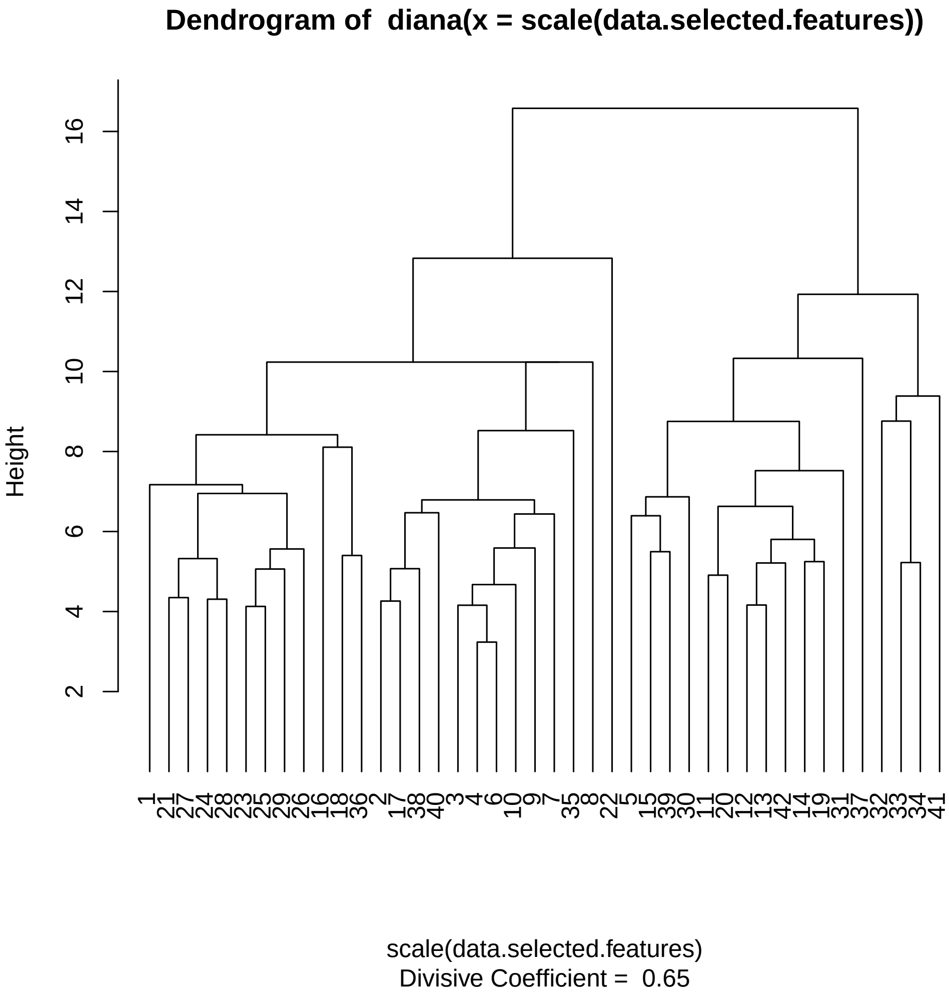

Figure 4.9. Dendrogram for DIANA clustering

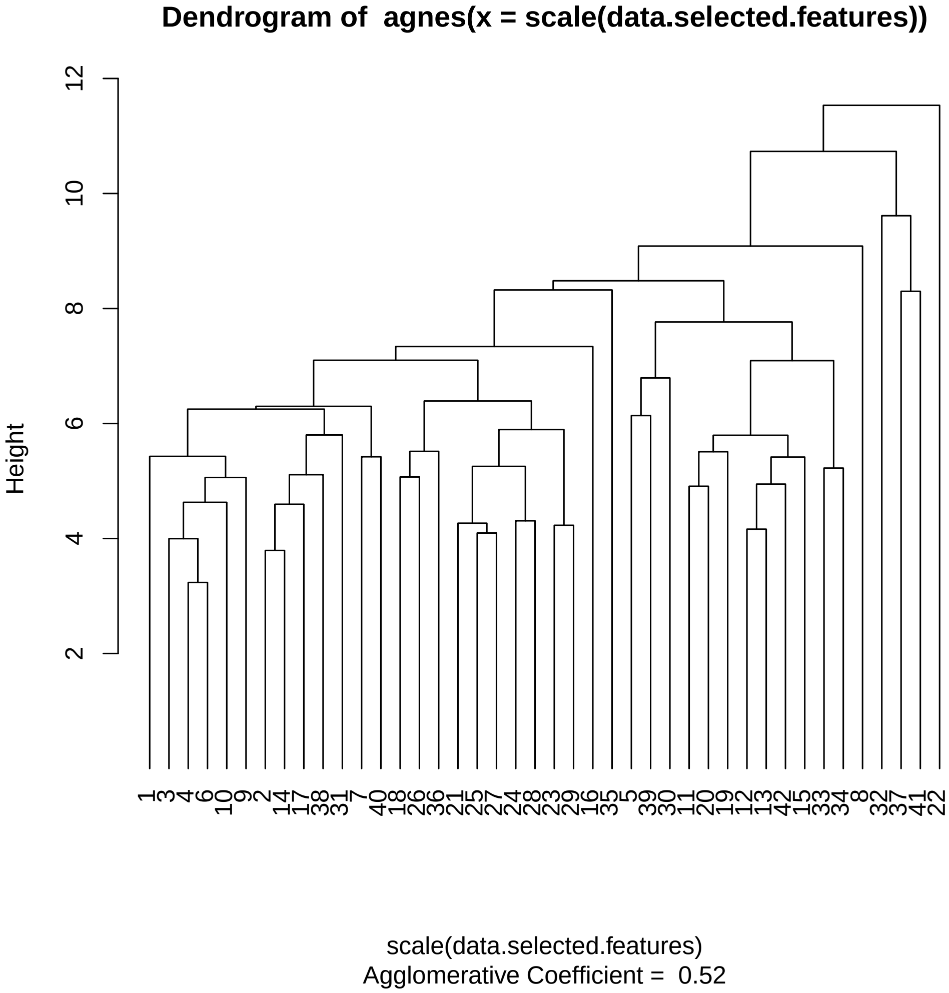

Figure 4.10. Dendrogram for AGNES clustering

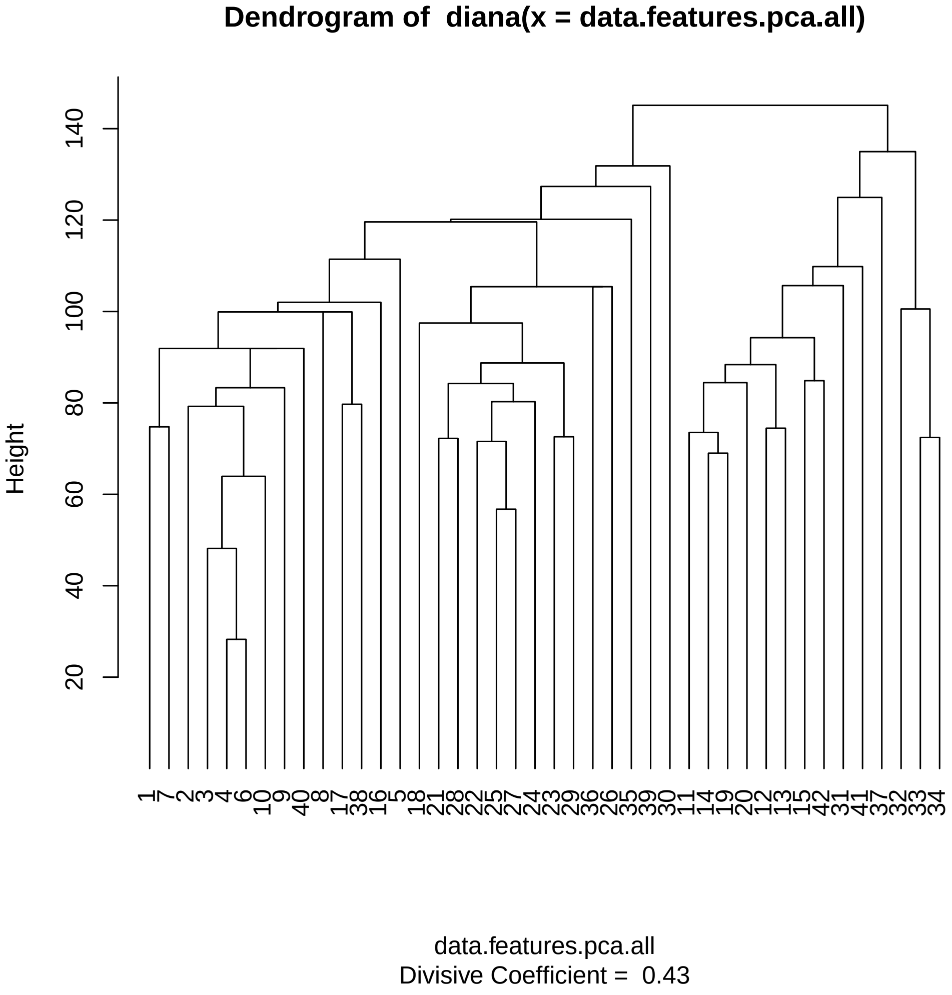

Figure 4.11. Dendrogram for DIANA clustering with PCA

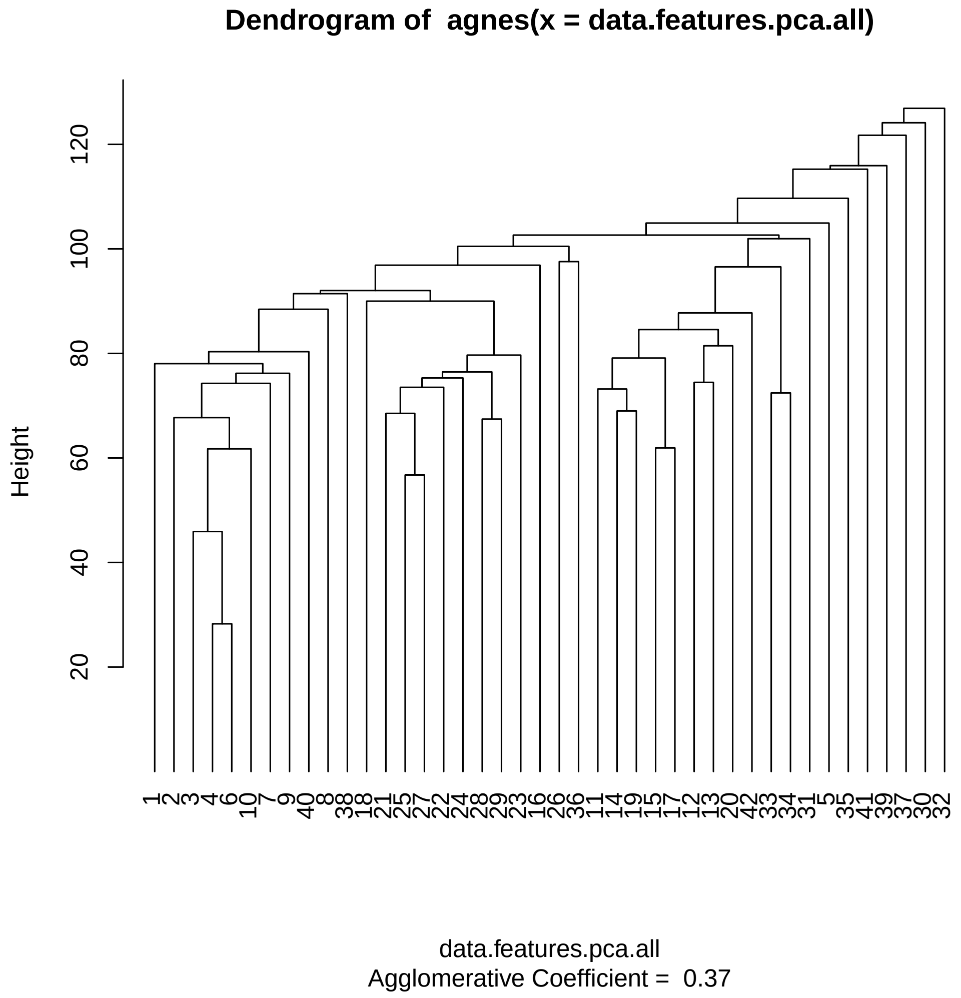

Figure 4.12. Dendrogram for AGNES clustering with PCA

5. Conclusions

Based on our research findings, we conclude that there is not a universal trend regarding the performance of our chosen classification algorithms with feature selection or dimensionality reduction. Analysis of test errors reveals that the PCA approach outperformed feature selection for kNN and Multinomial Logistic Regression, while Decision Tree exhibited its best performance with feature selection. Random Forest got identical results in terms of both approaches, but tuned parameters were different. It emphasises the need for individual development for each of these algorithms since they might be sensitive to different features manipulations.

Addressing the class imbalance issue through macro-averaged F1 scores yielded generally satisfactory results. However, in the cases of Decision Tree with PCA or Multinomial Logistic Regression with randomly chosen features, these models failed to predict one of our classes. This is a significant drawback that can be investigated in further research.

Overall, the most successful method was Multinomial Logistic Regression with the PCA approach. It demonstrated excellent accuracy and sensitivity to our class imbalance problem. Although K Nearest Neighbors algorithm with k=1 with PCA and Multinomial Logistic Regression with feature selection produced slightly higher errors, they still stand out as very capable models. As for LDA and QDA, their suitability depends on the number of features that we end up with.

When evaluating clustering techniques, we contrasted the unsupervised feature selection approach using the PAM algorithm with dimensionality reduction. Our analysis revealed that the choice of approach significantly influences the results, as evidenced by the comparison of internal validation indices. Furthermore, we examined how well the clustering results aligned with the actual classes. Interestingly, we found that the clusters do not always accurately reflect the actual classes. However, this was not the case for the PAM algorithm, which demonstrated a notable alignment between the clusters and the actual classes.

6. Further research suggestions

Indeed, our exploration of this multiclass classification problem has revealed numerous approaches, each with its own set of advantages and drawbacks. For the next phase of this project, it would be beneficial to explore additional algorithms such as XGBoost, neural networks, or alternative clustering algorithms that were not utilized in our current analysis. Additionally, conducting research focused on specific tumor types could shed light on whether certain types are notably more challenging to classify.

In summary, our project offers a meticulously crafted, data-driven perspective on this complex medical issue. By considering various methodologies and rigorously evaluating their performance, we aim to contribute valuable insights to the field.