Iga Świtalska, Mariia Kohut

Hepatisis data analysis

February 9, 2024

1. Data characteristics

We chose data: B) Medical diagnostics: Hepatitis Data Set. http://archive.ics.uci.edu/ml/datasets/Hepatitis

The main aim of our project is to analyze the dataset with clinical trial results of people with hepatitis and try to evaluate death risk. Hepatitis is a serious disease, inflammation of the liver from any cause, and it can lead to the death of a person. We want to find better diagnostic methods that should help to determine the risk of death due to hepatitis.

Our data set consists of 155 observations and 20 columns. The features are the following:

1. **Class** — a factor at two levels, which we want to predict (1 — patient dies, 2 — patient lives).
2. **Age** — age of the patients in years (from 20 to 80 years).
3. **Sex** — gender of the patient, a factor at two levels, coded by 1 (male) and 2 (female).
4. **Steroid** — steroid treatment, a factor at two levels coded by 1 (yes) and 2 (no).
5. **Antivirals** — antivirals medication, a factor at two levels, 1 (yes) and 2 (no).
6. **Fatigue** — fatigue is a frequent and disabling symptom reported by patients with chronic hepatitis, a factor at two levels 1 (yes) and 2 (no).
7. **Malaise** — malaise, one of the symptoms of hepatitis, is a factor at two levels, 1 (yes) and 2 (no).
8. **Anorexia** — anorexia, loss of appetite, a factor at two levels 1 (yes) and 2 (no).
9. **LiverBig** — the size of the liver increased or fatty, a factor at two levels 1 (yes) and 2 (no).
10. **LiverFirm** — the liver is firm, a factor at two levels, 1 (yes) and 2 (no).
11. **SpleenPalpable** — splenomegaly is an enlargement of the spleen, a factor at two levels 1 (yes) and 2 (no).
12. **Spiders** — enlarged blood vessels that resemble little spiders, a factor at two levels, 1 (yes) and 2 (no).
13. **Ascites** — ascites is the presence of excess fluid in the peritoneal cavity, a factor at two levels 1 (yes) and 2 (no).
14. **Varices** — varicose veins are a medical condition in which superficial veins become enlarged and twisted, a factor at two levels 1 (yes) and 2 (no).
15. **Bilirubin** — bilirubin is a substance made when the body breaks down old red blood cells.
16. **AlkPhosphate** — alkaline phosphatase is an enzyme made in liver cells and bile ducts, a discrete-valued feature reveals the level of alkaline phos-

phatase measured in IU/L, where UI — international unit. A 2013 research review showed that the normal range for a serum ALP level in healthy adults is 20 to 140 IU/L.

17. **Sgot** — a glutamic-oxaloacetic transaminase (SGOT) test measures the levels of the enzyme AST in the blood to assess liver health. A discrete valued feature measured in units per liter of serum. If the results of your SGOT test are high, that means one of the organs or muscles containing the enzyme could be damaged. The normal range of an SGOT test is generally between 8 and 45 units per liter of serum.
18. **Albumin** — albumin is a family of globular proteins, the most common of which is serum albumin. Low albumin levels can indicate a disorder of the liver or kidneys.
19. **Protime** — a discrete valued feature. How long it takes blood to form a clot in sec. It shows how a bad liver works.
20. **Histology** — histology is the branch of biology that studies the microscopic anatomy of biological tissues, a factor at two levels, 1 (yes) and 2 (no).

The first 6 rows of our data frame can be seen in the tables 1.1 and 1.2.

	Class	Age	Sex	Steroid	Antivirals	Fatigue	Malaise	Anorexia	LiverBig	LiverFirm
1	2	30	2	1	2	2	2	2	1	2
2	2	50	1	1	2	1	2	2	1	2
3	2	78	1	2	2	1	2	2	2	2
4	2	31	1	?	1	2	2	2	2	2
5	2	34	1	2	2	2	2	2	2	2
6	2	34	1	2	2	2	2	2	2	2

Table 1.1: First 6 observations

	SpleenPalpable	Spiders	Ascites	Varices	Bilirubin	AlkPhosphate	Sgot	Albumin	Protime	Histology
1	2	2	2	2	1.00	85	18	4.0	?	1
2	2	2	2	2	0.90	135	42	3.5	?	1
3	2	2	2	2	0.70	96	32	4.0	?	1
4	2	2	2	2	0.70	46	52	4.0	80	1
5	2	2	2	2	1.00	?	200	4.0	?	1
6	2	2	2	2	0.90	95	28	4.0	75	1

Table 1.2: First 6 observations

2. Clustering

As we mentioned before, we have a lot of missing values, and we should impute them. We found that the knn imputing method gave better results during the last part, so we will use it now.

There are 2 classes in the initial data, but we will not use this information in clustering and will try to determine the optimal number of clusters for each method. Moreover, we have mixed data, both numerical and categorical, so we use Gower's dissimilarity measure to calculate the dissimilarity matrix.

Let's visualize the dissimilarity matrix after ordering (figure 2.1). We can see one big blue group and the others are small.

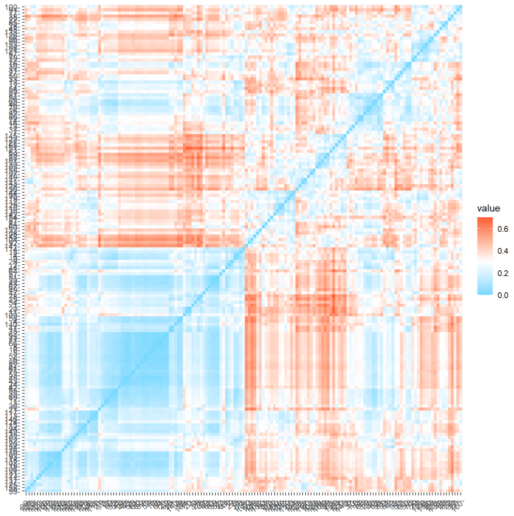


Figure 2.1: Dissimilarity matrix.

2.1. K-means

It is not possible to use the K-means method with mixed data, so we will use it only with numerical one. Also, we standardized our data. Since it is a partitioning cluster method, we first have to select the number of clusters (figure 2.2).

So, the optimal number of clusters for the K-means method:

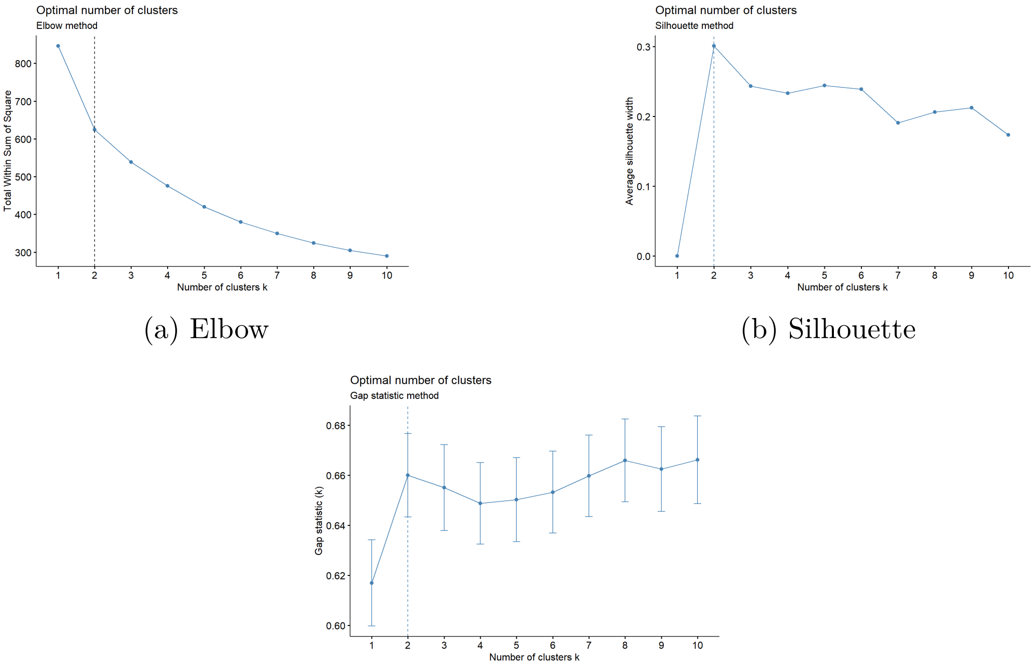
- Elbow method: it is difficult to determine the optimal number of clusters because it is hard to say if it seems like the bend in the knee, but the possible choice is 2;
- Silhouette method: it says that the optimal number of clusters is 2;
- Gap statistic method: it says that the optimal number of clusters is 2.

It's hard to say for sure how many clusters are in the dataset just looking at these statistics, but we can use 2 clusters in the K-means method.

Furthermore, we can run NbClust which computes up to 30 indices for determining the optimum number of clusters in a dataset and then takes a majority vote among them to see which is the optimum number of clusters (figure 2.3). According to the majority rule, the best number of clusters for the K-means method is 2, so we will use this one.

Since we use only numerical attributes here, we can use fviz_cluster to plot our clusters using PCA (use PCA only for visualization) (figure 2.4). PCA does not explain the data well, but the clusters are well separated in this plot.

The last part of the analysis is to validate the clusters found (figure 2.5). We can see, that the size of one cluster is 2 times larger than the other. The smaller cluster has a very small silhouette width, only 0.07. The average silhouette width is 0.3, which is not so good. Maybe other methods will be better. Now, let's compare our clusters with the original classes. The rand index is equal to 0.66, which is also not so good result.



(a) Elbow

(b) Silhouette

(c) Gap statistic

Figure 2.2: Optimal number of clusters for K-means method.

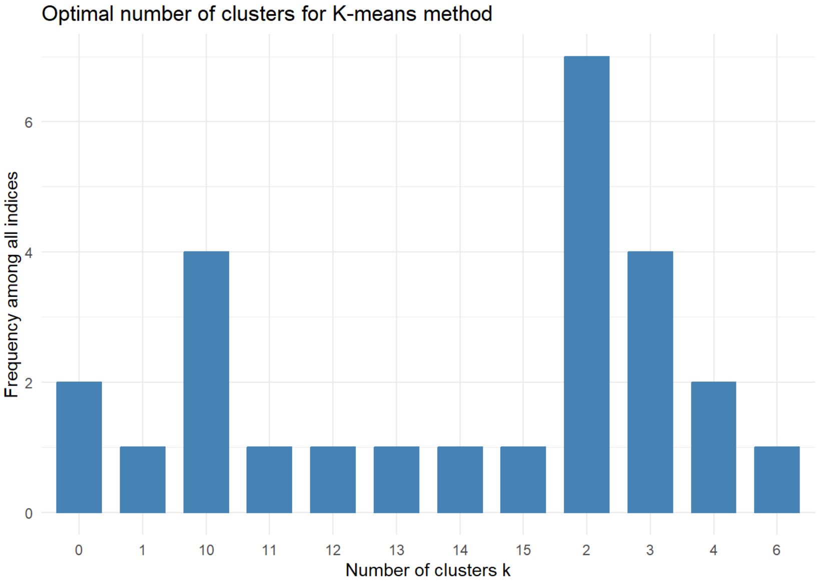


Figure 2.3: Optimal number of clusters for K-means method.

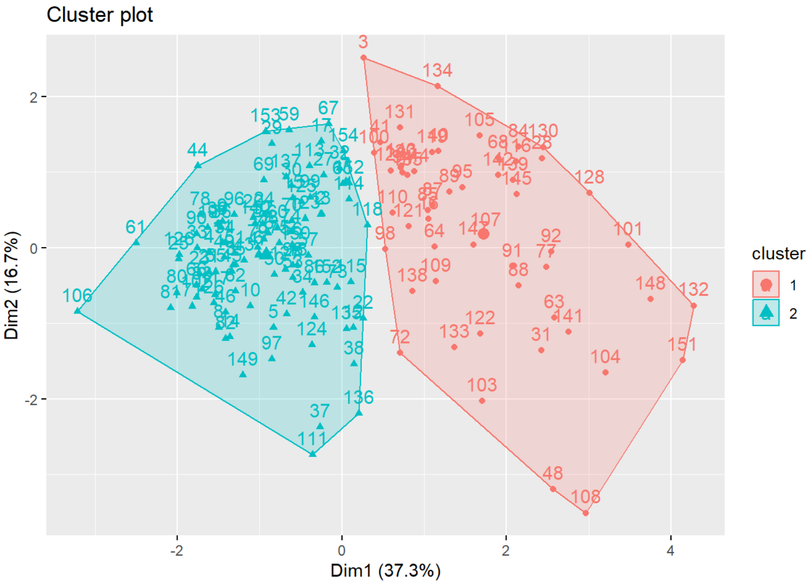


Figure 2.4: Visualization for K-means method.

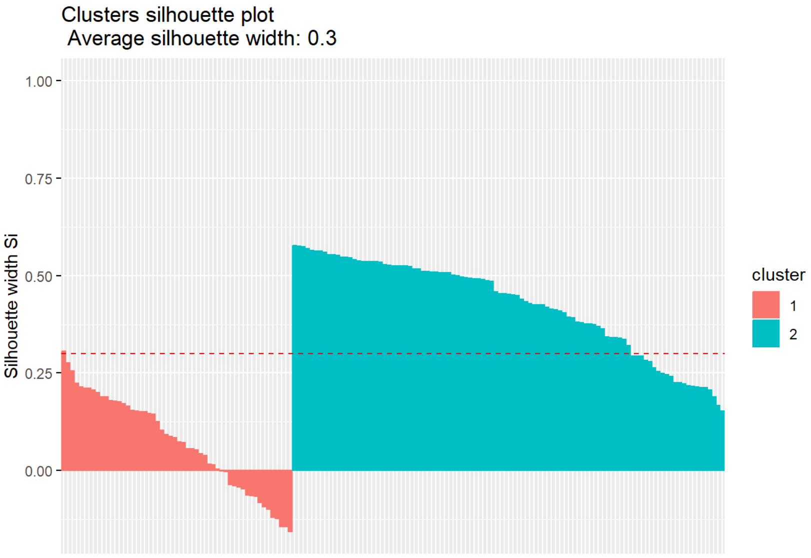


Figure 2.5: Silhouette index for K-means method.

2.2. Partition Around Medoids

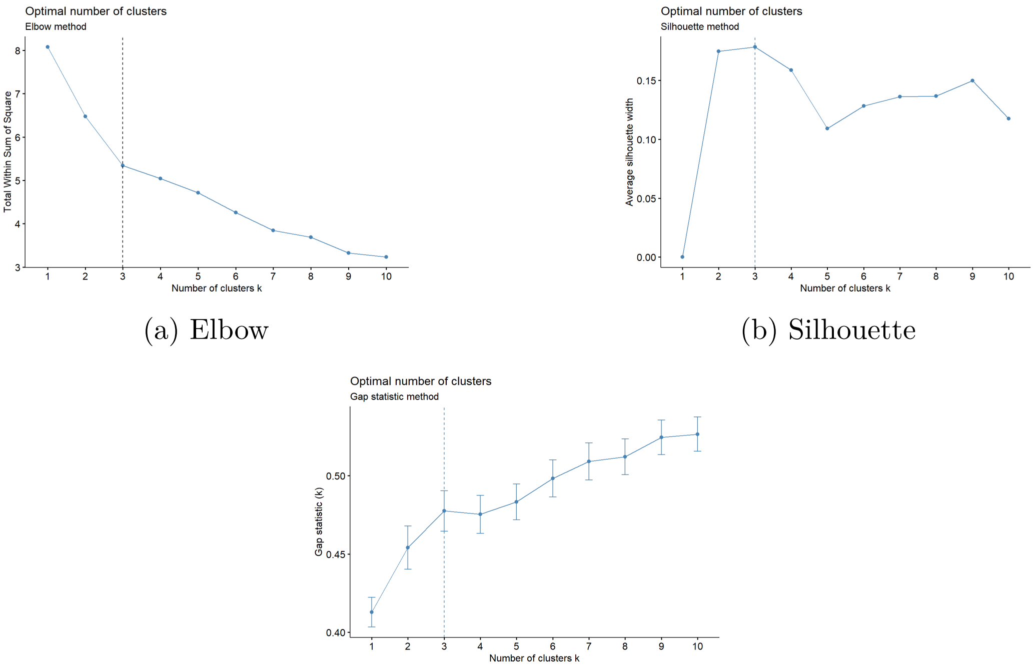
The next method is Partition Around Medoids. PAM can be used with mixed data, and it is less sensitive to outliers. Since it is a partitioning cluster method, we first have to select the number of clusters (figure 2.6).

So, the optimal number of clusters for the PAM method:

- Elbow method: it seems like 3 clusters because here it looks like a bend in the knee;
- Silhouette method: it says that the optimal number of clusters is 3;
- Gap statistic method: it also says that the optimal number of clusters is 3.

It's hard to say for sure how many clusters are in the dataset just looking at these statistics, but we can use 3 clusters in the PAM method. But we can also try 2 clusters.

The last part of the analysis is to validate the clusters found (figure 2.7). We can see, that for 2 clusters, the size of clusters is similar, but for 3 clusters, the size of one cluster is almost 2 times larger than the others. The average silhouette width is 0.28 and 0.25 for 2 and 3 clusters, respectively. For 2 clusters the result is slightly better, but it is still a poor one and worse than for the K-means method. Now, let's compare our clusters (for one where we use 2 clusters in the method) with the original classes. The rand index is equal to 0.59, which is also not so good result and less than for the K-means method. To sum up, the K-means method works better than Partition Around Medoids.

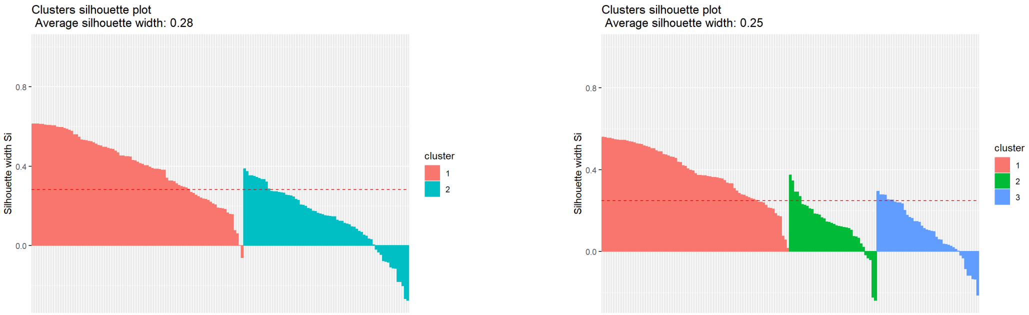


(a) Elbow

(b) Silhouette

(c) Gap statistic

Figure 2.6: Optimal number of clusters for PAM method.



(a) 2 clusters

(b) 3 clusters

Figure 2.7: Silhouette index for PAM method.

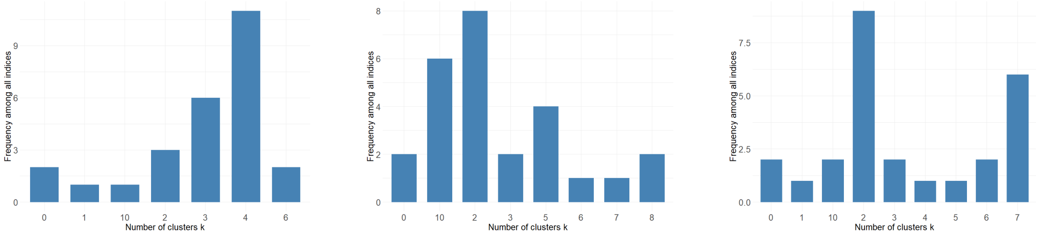
2.3. AGNES

We will select the optimal number of clusters based on the different indices. The optimal number of clusters will be the one selected by the largest number of (figure 2.8). It can be seen that for single and average methods, the preferred number of clusters is 2.

Now we will consider hierarchical methods. First, we will perform the AGNES algorithm for three linkage methods: complete, single, and average. We present AGNES clustering for all features and different linkage methods on the figures 2.9, 2.10, 2.11. It can be seen that the single linkage method performs poorly by assigning all observations to one cluster.

We will compare results for external and internal indices. The table 2.1 presents partitioning agreement between all the methods. The interpretation is similar to the accuracy in classification and, as with classification, we must be careful. Based on the table 2.1, we could incorrectly read that the single linkage method is almost as good as average linkage. However, knowing that we have a class imbalance problem and looking at the dendrogram, we can see that the single clustering method is actually the worst by assigning only one observation to the second cluster. Reading the rand index from table 2.2, we can draw similar conclusions. Additionally, we can observe that complete and average linkage methods are similar, and partition agreement is above 0.9.

We will now compare the average Silhouette width for all the linkage methods. From the charts 2.12 we can conclude that the AGNES method works poorly, especially for single linkage.



(a) Complete

(b) Single

(c) Average

Figure 2.8: Optimal number of clusters considering all variables.

	real	complete	single	average
real	1	0.77	0.80	0.82
complete	0.77	1	0.75	0.93
average	0.8	0.75	1	0.68
single	0.82	0.30	0.68	1

Table 2.1: Partition agreement.

	real	complete	single	average
real	1	0.65	0.68	0.70
complete	0.65	1	0.63	0.87
average	0.68	0.63	1	0.56
single	0.70	0.87	0.56	1

Table 2.2: Rand index.

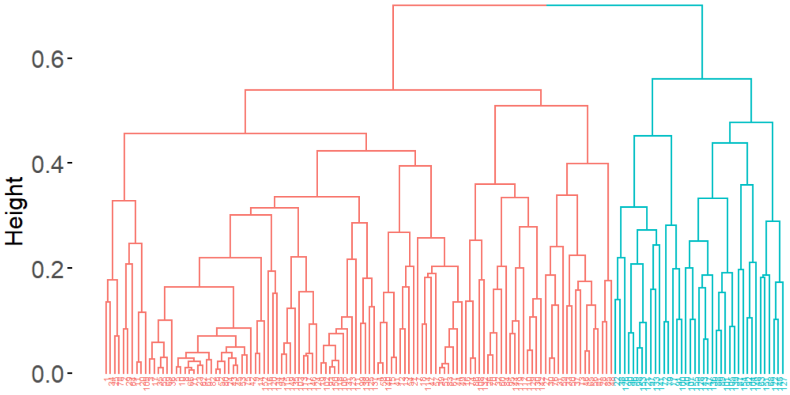


Figure 2.9: AGNES clustering for complete linkage.

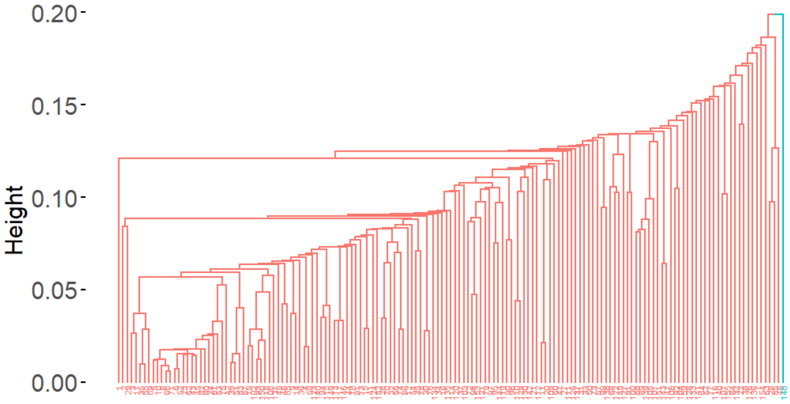


Figure 2.10: AGNES clustering for single linkage.

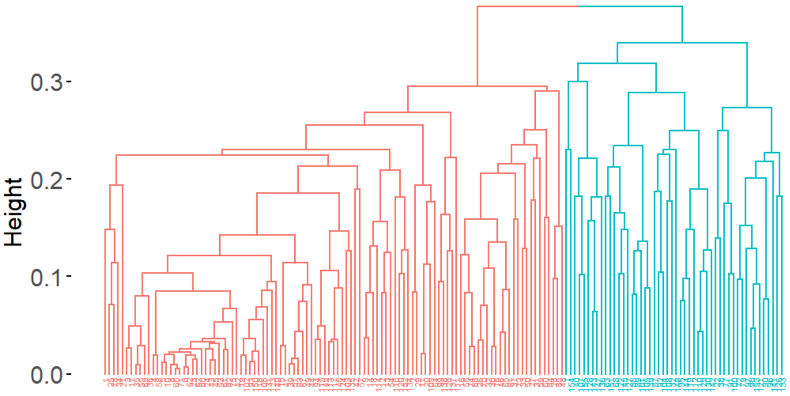
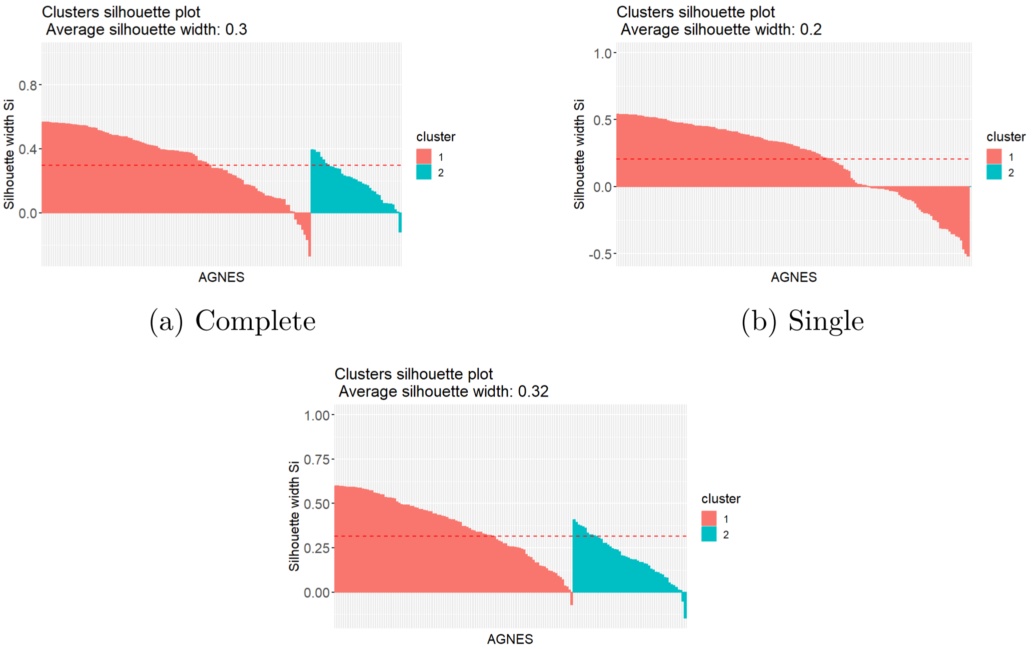


Figure 2.11: AGNES clustering for average linkage.



(a) Complete

(b) Single

(c) Average

Figure 2.12: Silhouette index for AGNES clustering.

2.4. DIANA

In the figure 2.13 we present the dendrogram for DIANA clustering. It looks similar to the AGNES method with an average linkage method. This observation can be confirmed by the Silhouette index presented in the figure 2.14. The partion agreement for DIANA is equal to 0.81 and the rand index is equal to 0.69. These results are again similar to the AGNES with the average linkage method.

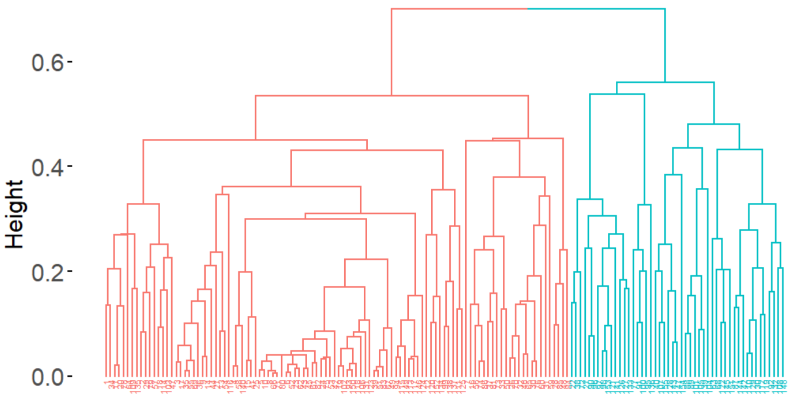


Figure 2.13: DIANA clustering.

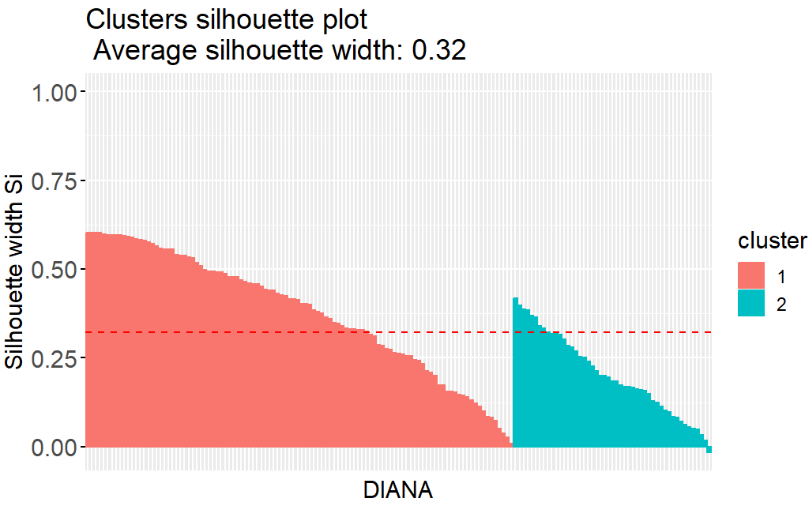


Figure 2.14: Silhouette index for DIANA clustering.

2.5. Fuzzy clustering

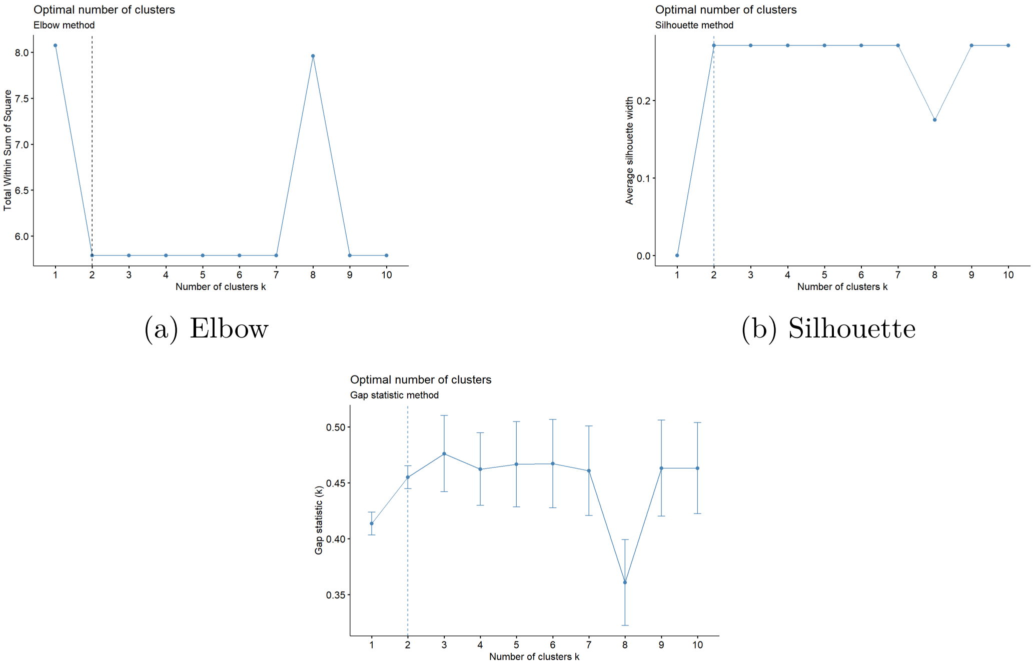
In fuzzy clustering, data points can potentially belong to multiple clusters. But it is a partitioning cluster method, and we first have to select the number of clusters (figure 2.15).

So, the optimal number of clusters for the Fuzzy clustering:

- Elbow method: it seems like 2 clusters because here it looks like a bend in the knee;
- Silhouette method: it says that the optimal number of clusters is 2;
- Gap statistic method: it also says that the optimal number of clusters is 2.

Just looking at these statistics, we can use 2 clusters in Fuzzy analysis.

The last part of the analysis is to validate the clusters found (figure 2.16). We can see, that the size of the clusters is similar. The silhouette width for clusters is very different: for one is 0.46, but for the other, it is critically small: 0.10. So the average silhouette width for data is just 0.29. Now, let's compare our clusters with the original classes. The rand index is equal to 0.56, which is also a bad result.



(a) Elbow

(b) Silhouette

(c) Gap statistic

Figure 2.15: Optimal number of clusters for Fuzzy method.

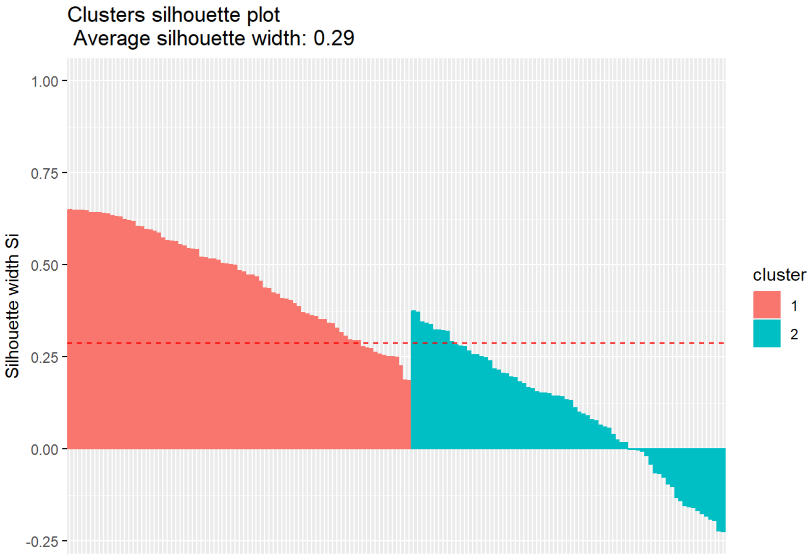


Figure 2.16: Silhouette index for Fuzzy method.

2.6. DBSCAN

Additionally, we tried the dbscan method using only continuous variables. The method is definitely not suitable for our clustering problem. Regardless of the choice of parameters, most observations are considered noise. Moreover, this method is very sensitive to parameter changes. Below are some of the results we received:

— eps = 0.5, minPts = 4: noise: 151, cluster1: 4,

— eps = 0.6, minPts = 4: noise: 126, cluster1: 6, cluster2: 13, cluster4: 5, cluster5: 5,

— eps = 0.7, minPts = 4: noise: 105, cluster1: 42, cluster2: 8,

— eps = 0.5, minPts = 4: noise: 151, cluster1: 4,

— eps = 0.6, minPts = 3: noise: 113, cluster1: 26, cluster2: 6, cluster4: 5, cluster5: 5,

— eps = 0.6, minPts = 5: noise: 139, cluster1: 11, cluster2: 5.

2.7. Comparison

Now, let's compare all clustering methods (table 2.3). For all methods, we used 2 clusters except PAM, where we also used 3 clusters, but the Silhouette index is smaller than for 2. We can see that the results are similar: clustering works badly for all methods. The worst is PAM and the best is DIANA and AGNES for average linkage, but they are still bad. We think it happened because one cluster is determined well, but another data overlapping and does not want to separate.

Method	Silhouette index	Rand index
K-means	0.30	0.66
PAM for 2 clusters	0.28	0.59
PAM for 3 clusters	0.25	—
AGNES for complete linkage	0.30	0.65
AGNES for single linkage	0.20	0.68
AGNES for average linkage	0.32	0.70
DIANA	0.32	0.69
Fuzzy clustering	0.29	0.56

Table 2.3: Comparison of clustering methods.

3. Dimensionality reduction

As we have both numerical and categorical attributes, we can not use the PCA (Principal Component Analysis) method. We decided to use MDS (Multidimensional Scaling). We use standardized data. Let us look at the scree plot 3.1.

The scree plot shows a clear elbow at dimension = 2, which suggests that a 2D solution should be adequate. Now we check out the Shepard diagram 3.2. The plot for $d = 2$ shows not so big amount of spread around the fitted function, like for $d = 1$ and there are not so crucial differences to one with d = 3. So, it is reasonable to use data with 2 dimensions.

So, we will use MDS for 2 dimensions (figure 3.3). The classes are separated but also overlap a lot, so in the future, we may have a problem with classification. To sum up, we have data with two numerical features and a target attribute. In the table 3.1 we present the summary of the new data. It can be observed that the mean of columns is 0.

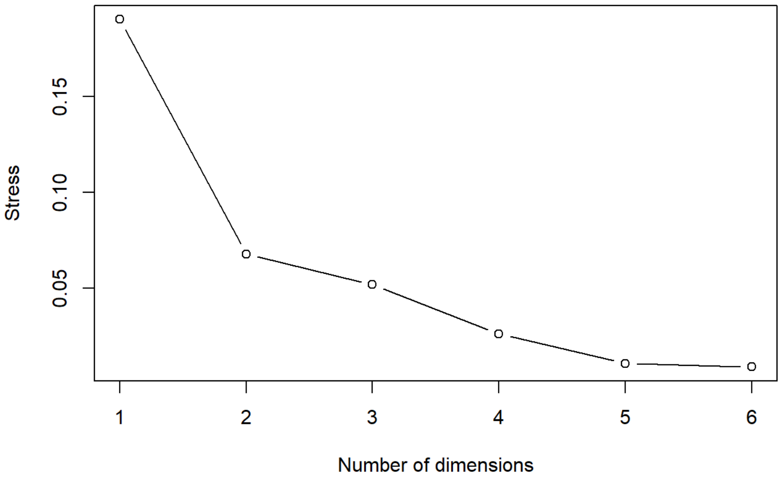


Figure 3.1: A scree plot.

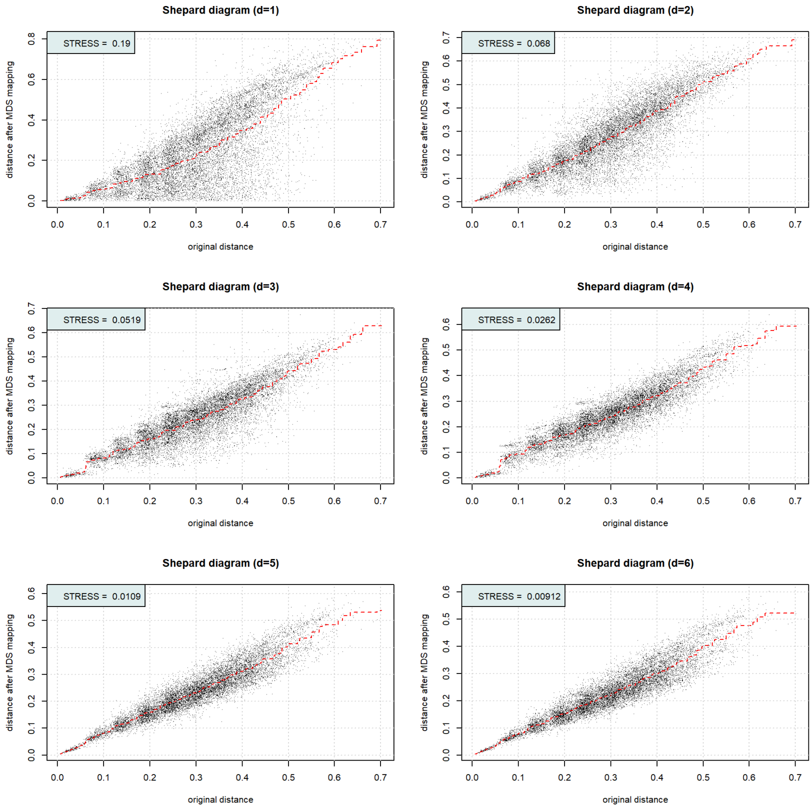


Figure 3.2: Shepard diagram.

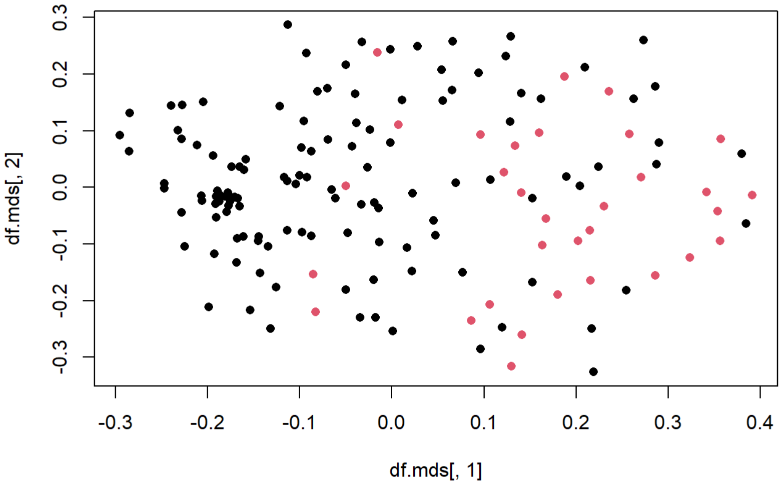


Figure 3.3: Visualization of MDS.

X1		X2		Class	
Min.	-0.29530	Min.	-0.325977	0	123
1st Qu.	-0.15952	1st Qu.	-0.088834	1	32
Median	-0.03263	Median	-0.008753		
Mean	0.00000	Mean	0.000000		
3rd Qu.	0.13709	3rd Qu.	0.093633		
Max.	0.39117	Max.	0.286861		

Table 3.1: Summary of new data.

3.1. Classification

We will now return to the classification and check how the results have changed after the dimensionality reduction. We first balance our training set using the MWMOTE oversampling technique. In the figure 3.4 we present our dataset before and after oversampling.

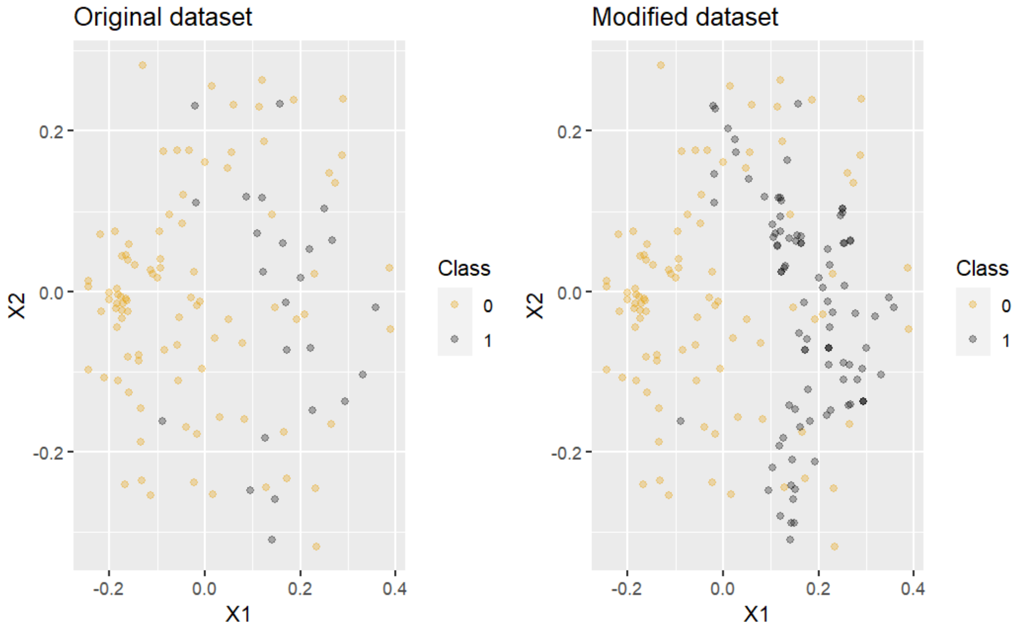


Figure 3.4: Comparison between starting data and after oversampling.

3.1.1. Linear regression

The first classification model is linear regression. In the figure 3.5 we present results obtained for the test set using two MDS components. In table 3.2 we additionally present the comparison between the previous and our new models. It can be seen that after dimensionality reduction we get better results than using all features before the transformation. The results after feature selection are a little better but still comparable.

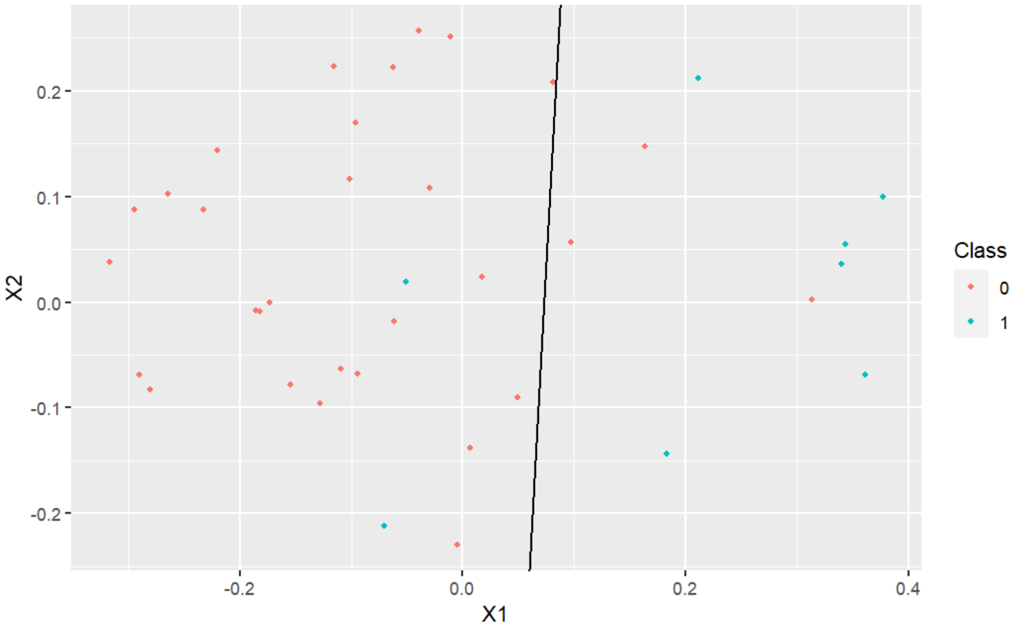


Figure 3.5: Linear regression after dimensionality reduction.

	Accuracy	Kappa	Sensitivity	Specificity	F1	Balanced Accuracy
MDS	0.87	0.62	0.75	0.90	0.71	0.83
All	0.84	0.56	0.75	0.87	0.67	0.81
Important	0.92	0.75	0.75	0.97	0.80	0.86
Protime + Bilirubin	0.89	0.68	0.75	0.93	0.75	0.84

Table 3.2: Metrics comparison for different linear regression models.

3.1.2. Logistic regression

In the figure 3.6 we present results obtained for the test set using two MDS components. We can see that predicted posterior probabilities are reasonable. There are false predictions in all models, but not so many. Let's compare our new results with the previous ones (table 3.3). As in the case of linear regression, the results for data after dimensionality reduction are similar to those after feature selection.

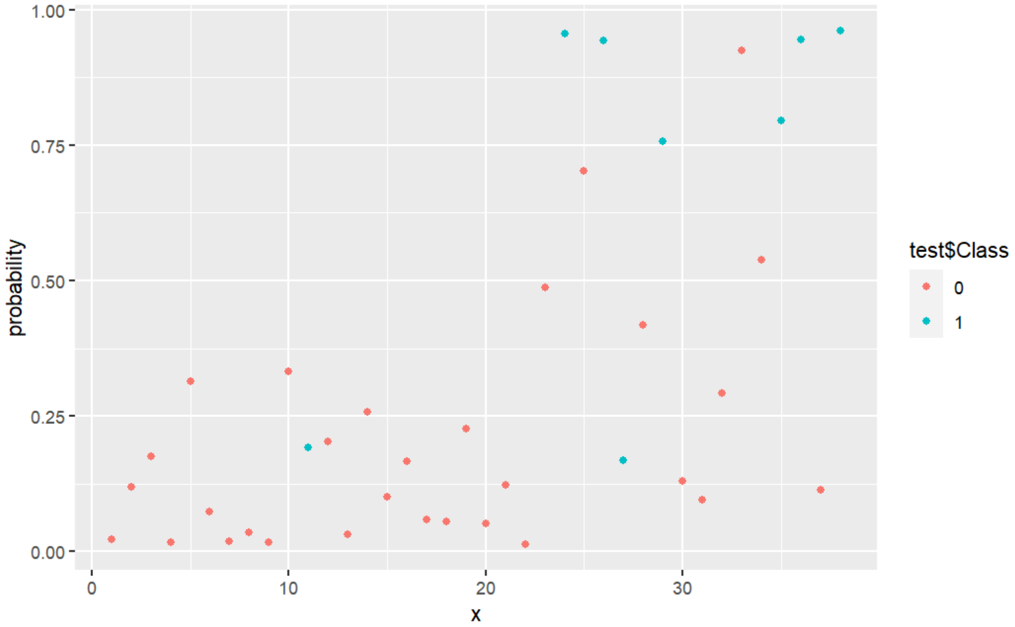


Figure 3.6: Logistic regression after dimensionality reduction.

	Accuracy	Kappa	Sensitivity	Specificity	F1	Balanced Accuracy
MDS	0.87	0.62	0.75	0.90	0.71	0.83
All	0.87	0.59	0.62	0.93	0.67	0.78
Important 1	0.92	0.75	0.75	0.97	0.80	0.86
Important 2	0.87	0.62	0.75	0.90	0.71	0.82

Table 3.3: Metrics comparison for different logistic regression models.

3.1.3. KNN

First, we performed 5 times repeated 5-fold Cross-Validation to choose the best number of neighbors. Based on figure 3.7 we choose the 5-NN model. Compared to previous models (table 3.4), our new model behaves similarly. It is slightly better in terms of sensitivity than the model using all variables but worse in terms of specificity. It is also worse than models using feature selection.

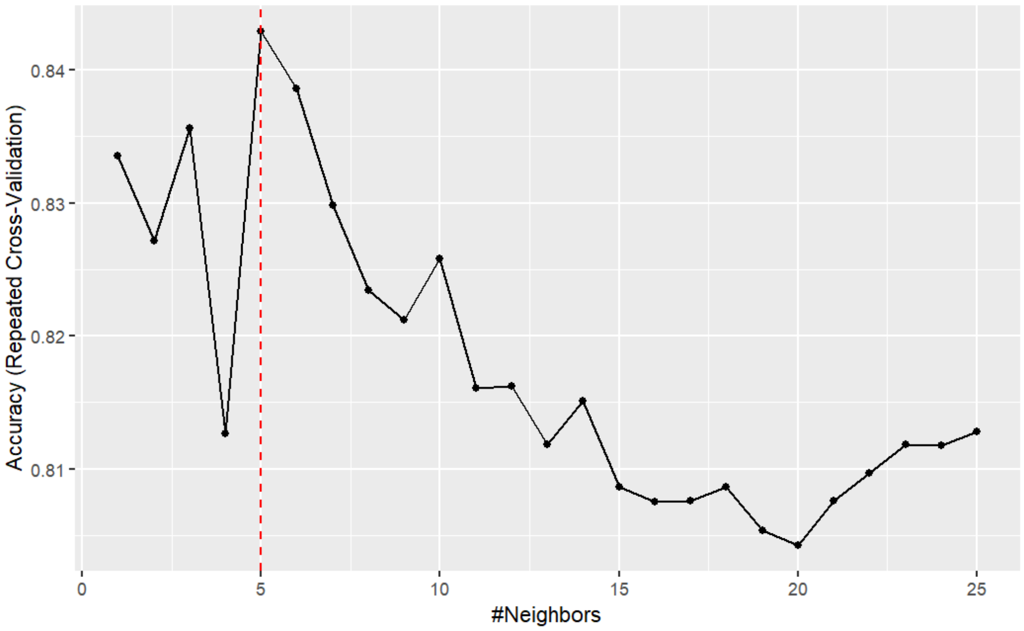


Figure 3.7: Accuracy based on the number of nearest neighbors.

	Accuracy	**Kappa**	**Sensitivity**	**Specificity**	**F1**	**Balanced Accuracy**	**k**
MDS	0.76	0.32	0.50	0.83	0.47	0.67	5
All	0.74	0.38	0.75	0.73	0.55	0.74	1
Important	0.79	0.46	0.75	0.80	0.60	0.78	7
Protime + Bilirubin	0.82	0.55	0.88	0.80	0.67	0.84	18

Table 3.4: Metrics comparison for different knn models.

3.1.4. Decision tree

Next we fitted decision tree model to the data after dimensionality reduction. We pruned our tree based on the 1-SE rule and the optimal number of splits was equal to 2. The results are presented on the figure 3.8. We also tried using 3 MDS components, which gives slightly better results (table 3.5). The second tree is shown in the figure 3.9. Based on the decision trees we may also conlude that the first dimension is more imporatant than the others

Based on the table 3.5 we can conclude one again, that feature selection gives better results than dimensionality reduction, even when increasing number of MDS components.

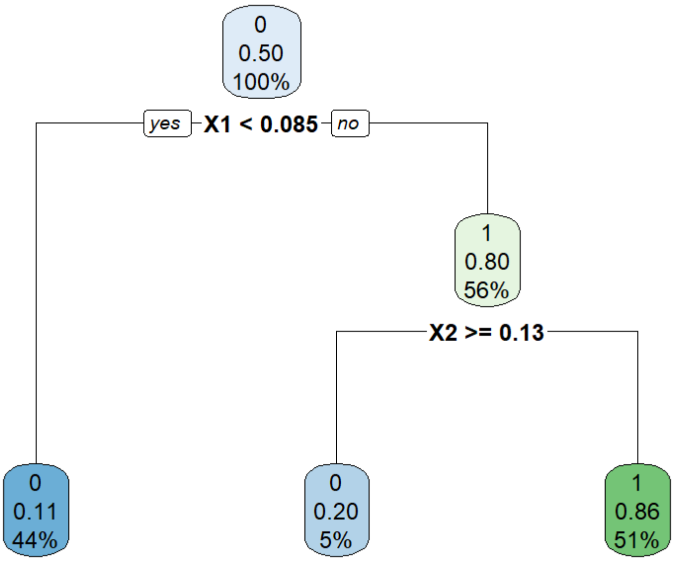


Figure 3.8: Decision tree after dimensionality rduction

	Accuracy	Kappa	Sensitivity	Specificity	F1	Balanced Accuracy
MDS (2)	0.87	0.59	0.63	0.93	0.67	0.87
MDS (3)	0.89	0.68	0.75	0.93	0.75	0.84
All	0.84	0.42	0.38	0.97	0.50	0.67
Important	0.92	0.72	0.62	1.00	0.77	0.81

Table 3.5: Metrics comparison for different decision tree models.

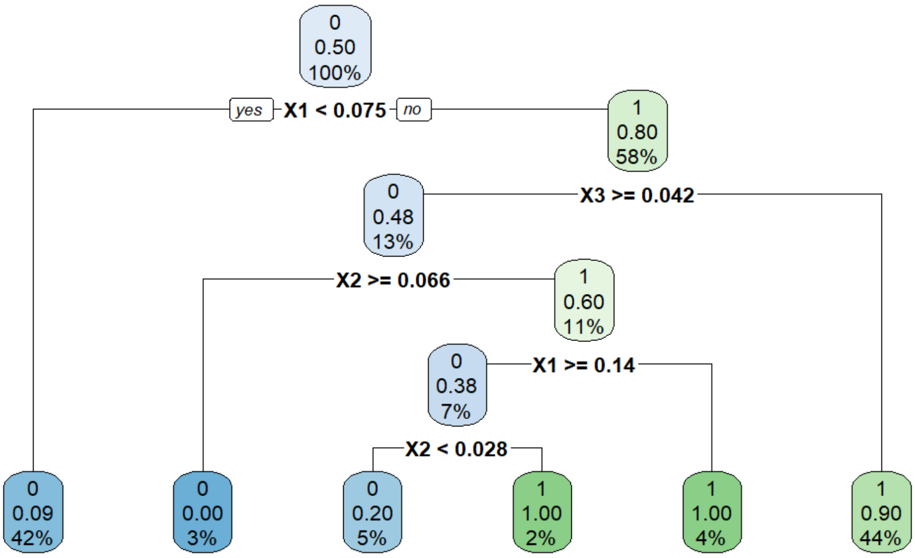


Figure 3.9: Decision tree after dimensionality rduction

3.1.5. Random forest

Finally we use random forest model, which have the best results in previous analysis. Again in the table 3.6 we can see that dimensionality reduction worsened our results. Even a model that uses all variables without transformations performs better.

	Accuracy	Kappa	Sensitivity	Specificity	F1	Balanced Accuracy
MDS (2)	0.82	0.42	0.50	0.90	0.53	0.70
MDS (3)	0.84	0.53	0.63	0.90	0.63	0.76
All	0.92	0.72	0.62	1.00	0.77	0.81
Important	0.92	0.72	0.62	1.00	0.77	0.81

Table 3.6: Metrics comparison for different random forest models.

3.1.6. Conclusion

In summary, we did not manage to improve the behavior of the models. Although models trained on data after dimensionality reduction perform better than those that use all variables, it is better to consider the feature selection for our data. Summary of all the resultes obtained after dimensionality reduction are presented in the table 3.7. Surprisingly the best models are linear and logistic regression. They have the same accuracy as decision tree, but higher sensitivity, which is important for our analysis. It refers to the probability that the model will predict death, provided that the patient actually died.

In the future we may also consider using three MDS components, as they give better results for decision tree and random forest.

	Accuracy	Kappa	Sensitivity	Specificity	F1	Balanced Accuracy
Linear regression	0.87	0.62	0.75	0.90	0.71	0.83
Logistic regression	0.87	0.62	0.75	0.90	0.71	0.83
KNN	0.76	0.32	0.50	0.83	0.47	0.67
Decision tree	0.87	0.59	0.63	0.93	0.67	0.87
Random forest	0.82	0.42	0.50	0.90	0.53	0.70

Table 3.7: Comparison of metrics for models trained on post-MDS data.

3.2. Clustering

As we mentioned before, we have two numerical features, so we use Euclidean distance for clustering. We standardized our data.

Let's visualize the dissimilarity matrix after ordering. The dissimilarity matrix 3.10 looks better than for initial data. We can see that more values near the diagonal are blue.

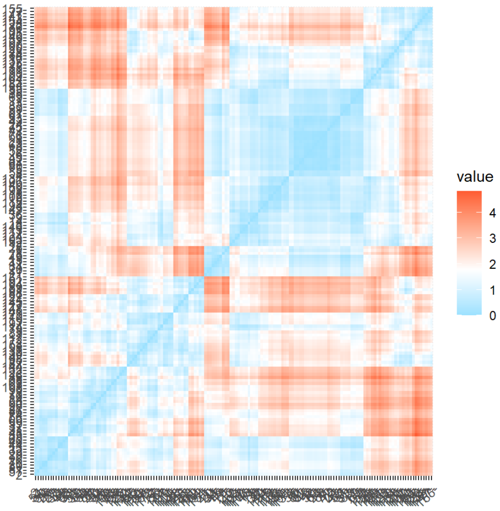


Figure 3.10: Dissimilarity matrix.

3.2.1. K-means

As we have only numerical data, we do not need to divide our dataset. Since it is a partitioning cluster method, we first have to select the number of clusters. Based on the figure 3.11, the optimal number of clusters for the K-means method:

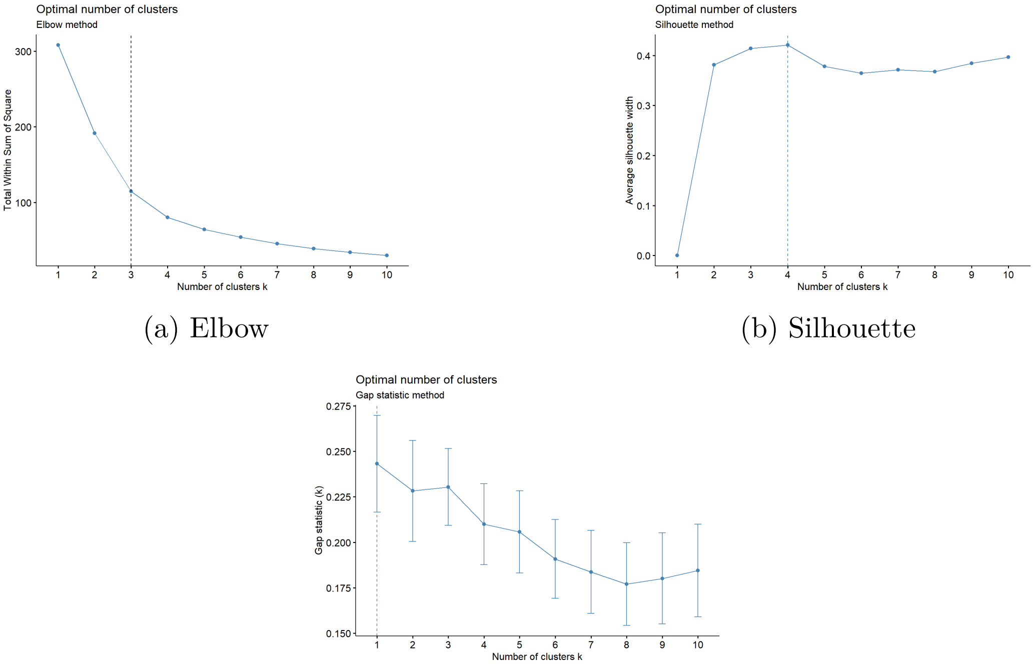
- Elbow method: it is difficult to determine the optimal number of clusters because it is hard to say if it seems like the bend in the knee, but the possible choice is 3;
- Silhouette method: it says that the optimal number of clusters is 4;
- Gap statistic method: it is strange, but it says that the optimal number of clusters is 1, however, we also see that we can use 3 clusters.

It's hard to say for sure how many clusters are in the dataset just looking at these statistics, but we can use 3 or 4 clusters in the K-means method. Furthermore, we run NbClust. The results are presented in the figure 3.12.

According to the majority rule, the best number of clusters for the K-means method is 2, 3, or 4, so we will use them. As we have only 2 attributes, we can use fviz_cluster to plot our clusters.

Now, let's visualize our clusters. We can see that the clusters are well separated for all number of clusters (figure 3.13). We can also see, that the size of one cluster is larger than the others for all number of clusters. As shown in the figure 3.14, the bigger the cluster is, the higher the silhouette width it has. The average silhouette width is 0.38, 0.41, and 0.42 for 2, 3, and 4 clusters, respectively. For 4 clusters it works better, we can also mention that the results are better than for initial data, but we compare it in more detail later.

Now, let's compare our clusters (for one where we use 2 clusters in the method) with the original classes. The rand index is equal to 0.58.



(a) Elbow

(b) Silhouette

(c) Gap statistic

Figure 3.11: Optimal number of clusters for K-means method.

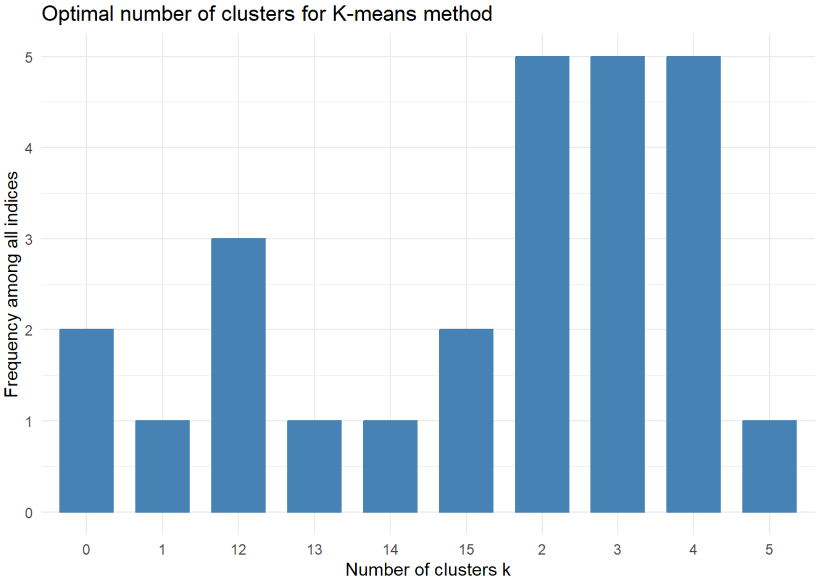
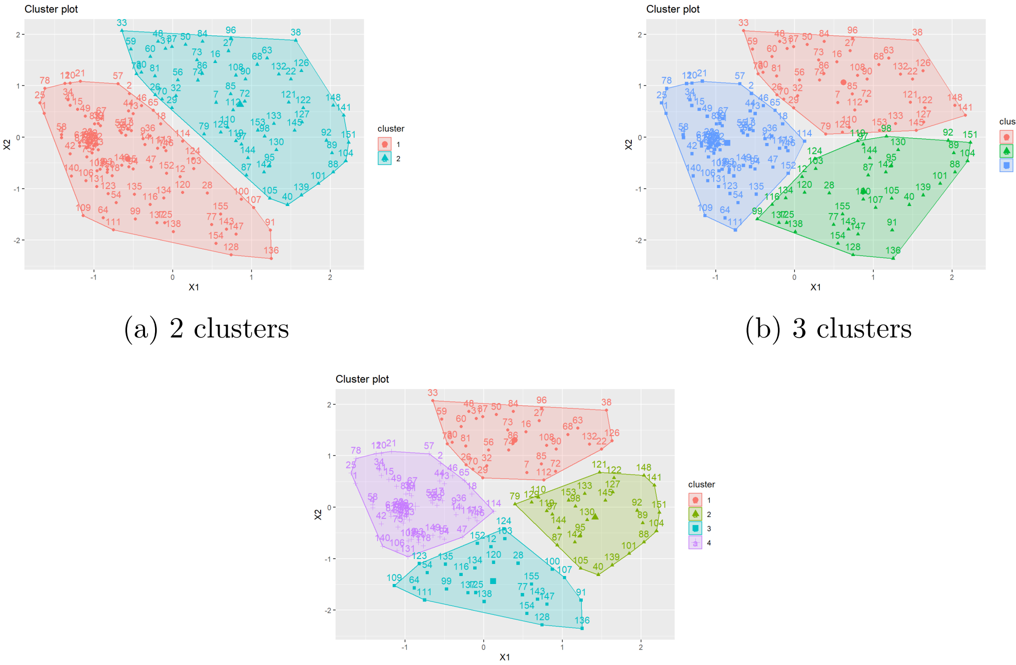


Figure 3.12: Optimal number of clusters for K-means method.

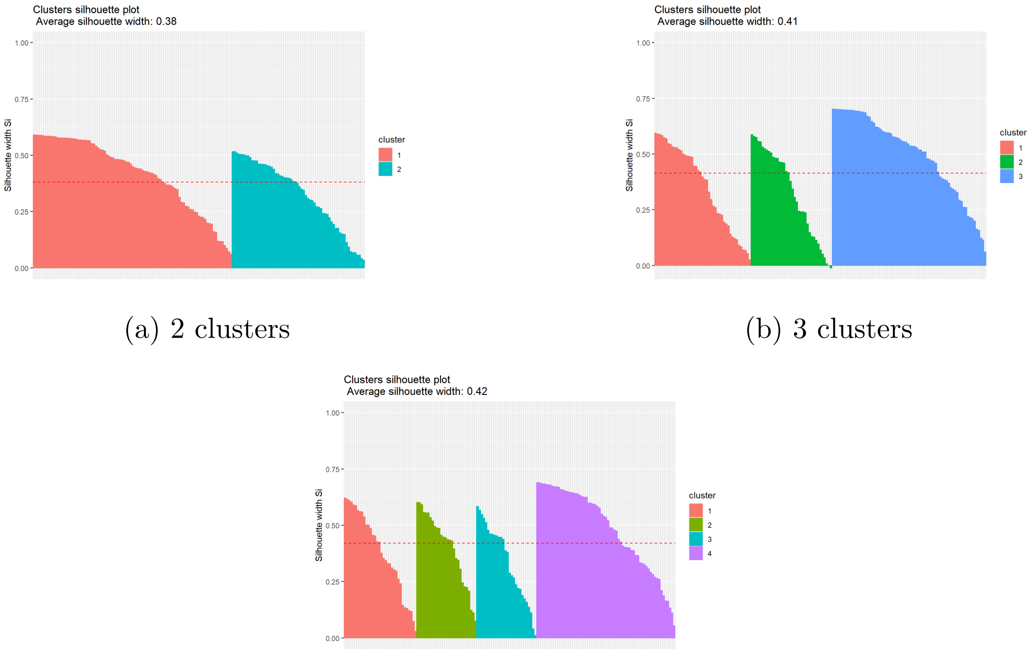


(a) 2 clusters

(b) 3 clusters

(c) 4 clusters

Figure 3.13: Visualization for K-means method.



(a) 2 clusters

(b) 3 clusters

(c) 4 clusters

Figure 3.14: Silhouette index for K-means method.

3.2.2. Partition Around Medoids

The next method is Partition Around Medoids. Since it is a partitioning cluster method, we first have to select the number of clusters. Based on the figure 3.15, the optimal number of clusters for the PAM method:

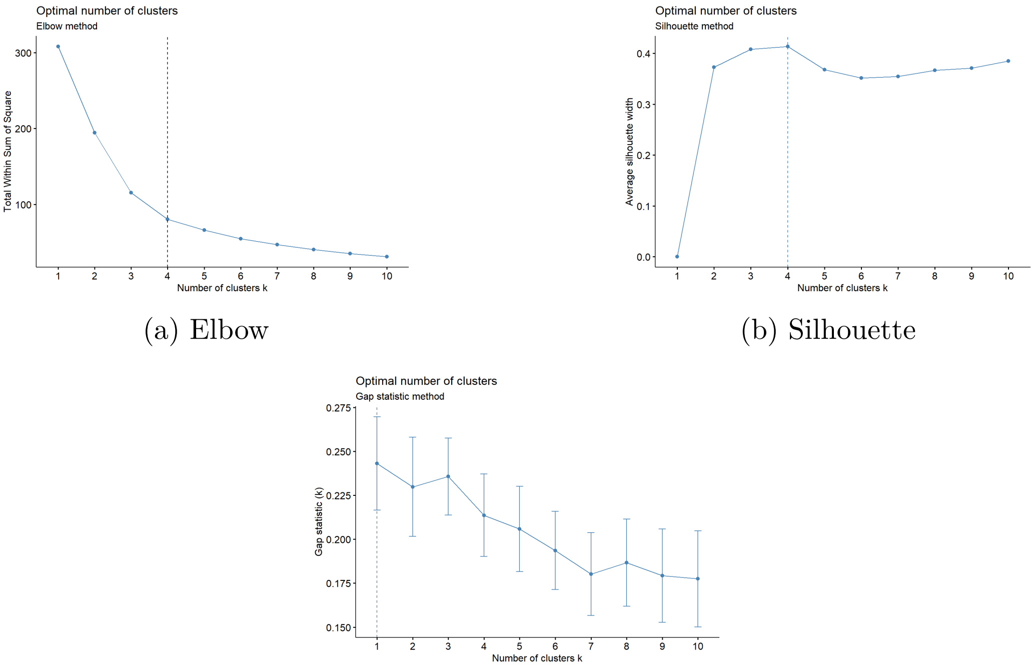
— Elbow method: it is hard to determine the optimal number of clusters because it is hard to say if it seems like the bend in the knee, but the possible choice is 4;

— Silhouette method: it says that the optimal number of clusters is 4;

— Gap statistic method: it says that the optimal number of clusters is 1, however, we also see that we can use 3 clusters.

It's hard to say for sure how many clusters are in the dataset just looking at these statistics, but we can use 3 and 4 clusters in the PAM method. But we can also try 2 clusters. Now, let's visualize our clusters. In the figure 3.16 we can see that the clusters are well separated for all number of clusters.

The last part of the analysis is to validate the clusters found. In the figure 3.17 it can be obsered, that the size of one cluster is larger than the others for all number of clusters. The average silhouette width is 0.37, 0.41, and 0.41 for 2, 3, and 4 clusters, respectively. For 2 clusters it works worse. We can also mention that results are better than for initial data. Now, let's compare our clusters with the original classes for 2 clusters. Now, let's compare our clusters (for one where we use 2 clusters in the method) with the original classes. The rand index is equal to 0.60.

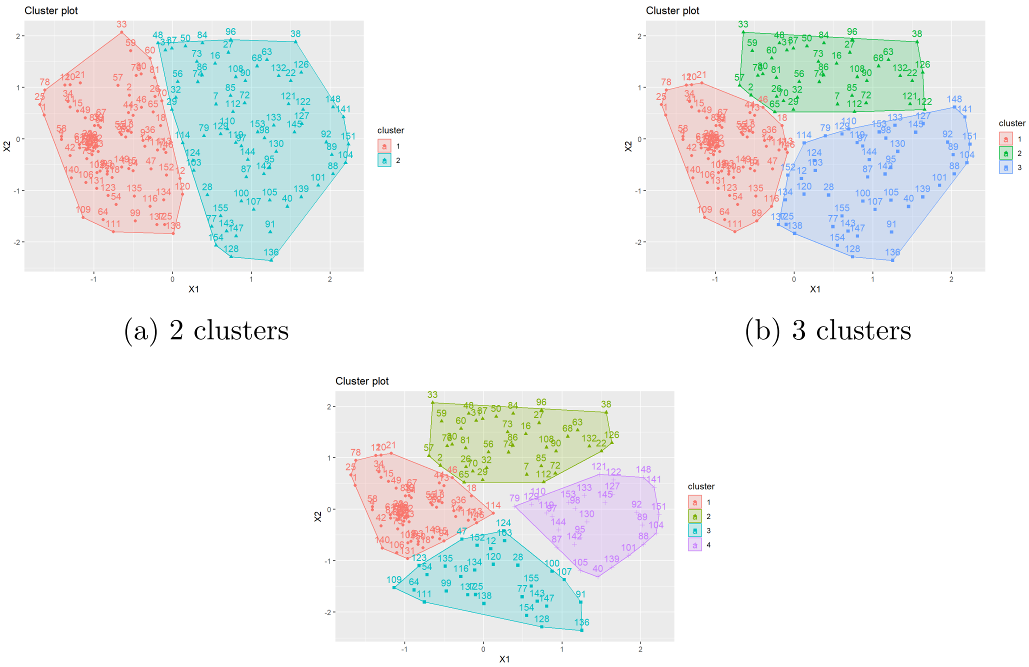


(a) Elbow

(b) Silhouette

(c) Gap statistic

Figure 3.15: Optimal number of clusters for PAM method.

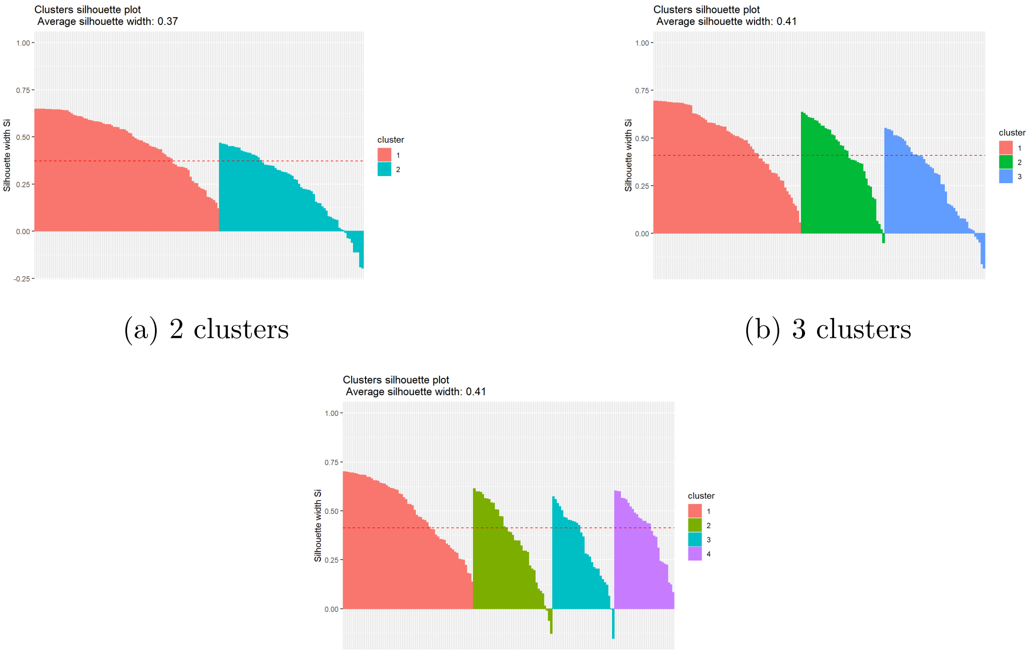


(a) 2 clusters

(b) 3 clusters

(c) 4 clusters

Figure 3.16: Visualization for PAM method.



(a) 2 clusters

(b) 3 clusters

(c) 4 clusters

Figure 3.17: Silhouette index for PAM method.

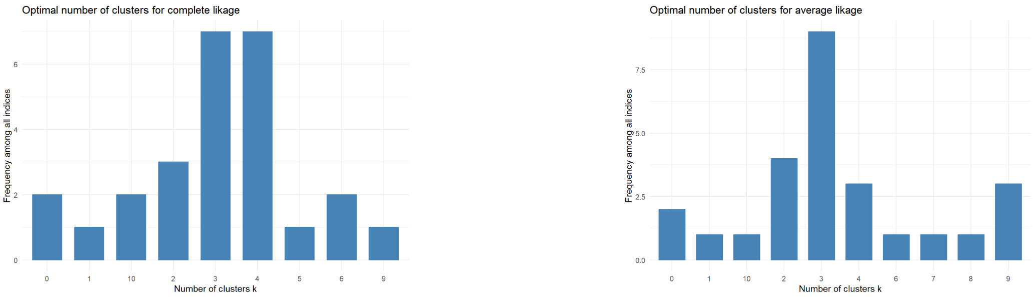
3.2.3. AGNES

Now we will consider hierarchical methods. During the previous clustering we saw, that it works badly for single linkage, so we will not use it now. We will select the optimal number of clusters based on the different indices. For complete linkage we will use 3 and 4 clusters, and for average linkage only 3 clusters. In addition, we will use 2 clusters in order to compare it with previous results.

We present AGNES clustering for all features and different linkage methods in the figures 3.19, 3.20. Now, let's visualize our clusters. In the figures 3.21 and 3.22 We can see that the clusters overlap, especially for average linkage with 2 clusters.

We will now compare the average Silhouette width for all the linkage methods. From the charts 3.23 and 3.24 we can conclude that the AGNES method works poorly. We can see, that the size of one cluster is larger than the others for all number of clusters. The average silhouette width for complete linkage is 0.35, 0.38, and 0.39 for 2, 3, and 4 clusters, respectively, and for average linkage is 0.28 and 0.35 for 2 and 3 clusters, respectively. For 2 clusters it works worse.

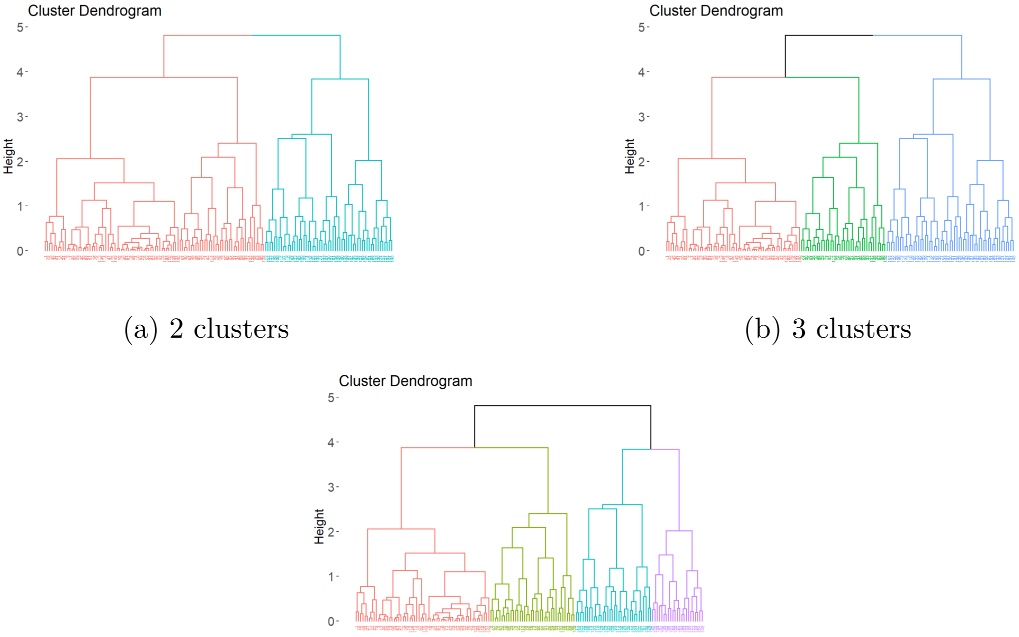
Now, let's compare our clusters (for one where we use 2 clusters in the method) with the original classes. For complete linkage, the rand index is equal to 0.61, and for average – 0.70.



(a) Complete

(b) Average

Figure 3.18: Optimal number of clusters considering all variables.

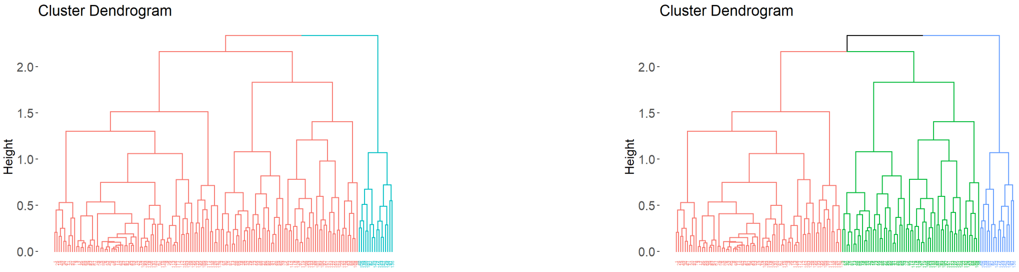


(a) 2 clusters

(b) 3 clusters

(c) 4 clusters

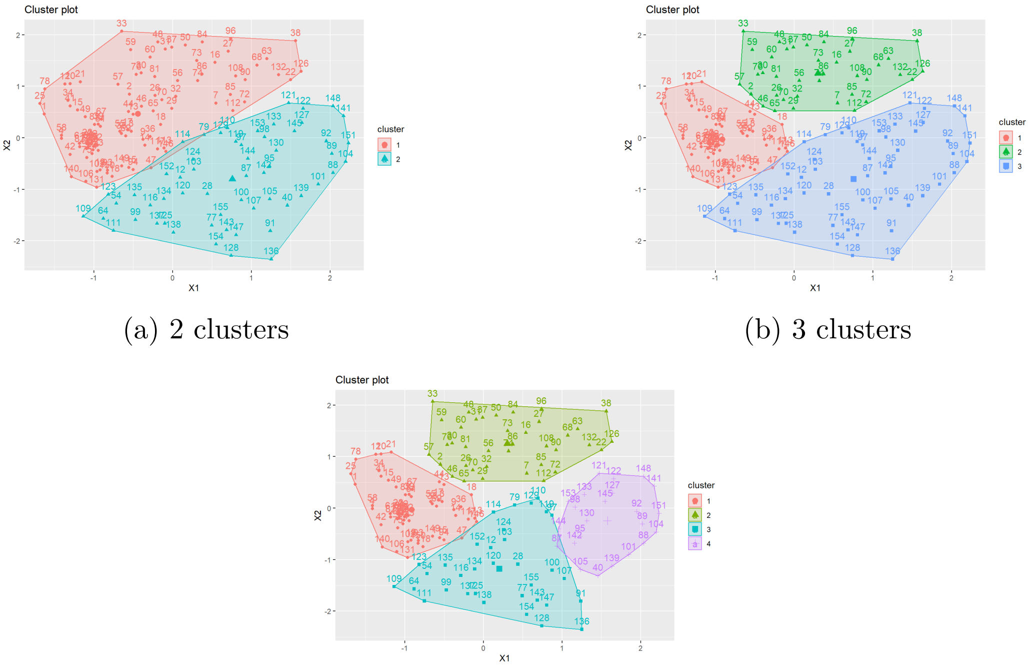
Figure 3.19: AGNES clustering for complete linkage.



(a) 2 clusters

(b) 3 clusters

Figure 3.20: AGNES clustering for average linkage.

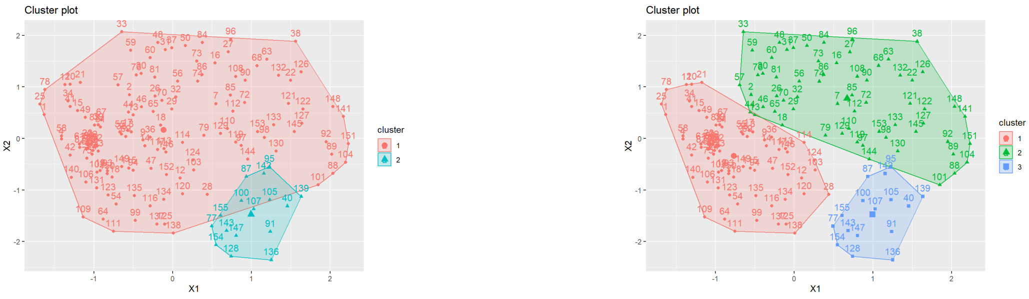


(a) 2 clusters

(b) 3 clusters

(c) 4 clusters

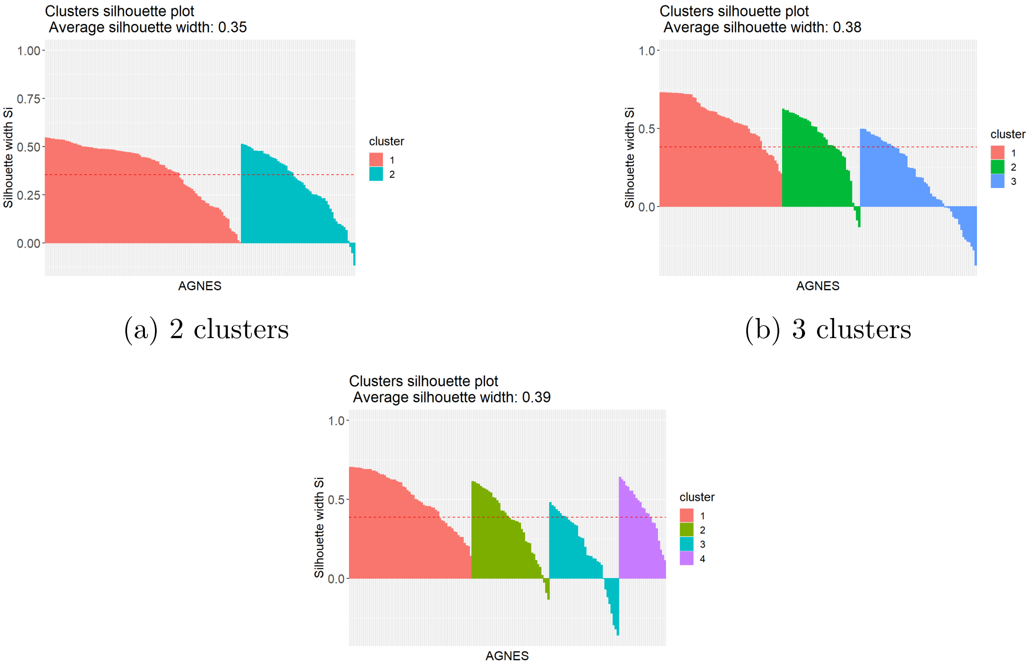
Figure 3.21: Visualization for AGNES clustering for complete linkage.



(a) 2 clusters

(b) 3 clusters

Figure 3.22: Visualization for AGNES clustering for average linkage.

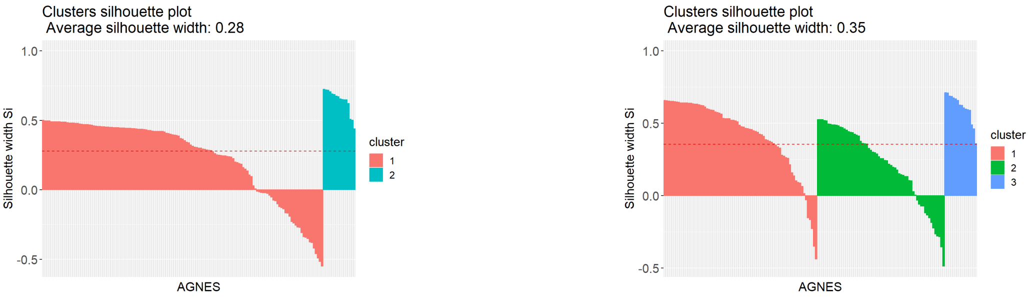


(a) 2 clusters

(b) 3 clusters

(c) 4 clusters

Figure 3.23: Silhouette index for AGNES clustering for complete linkage.



(a) 2 clusters

(b) 3 clusters

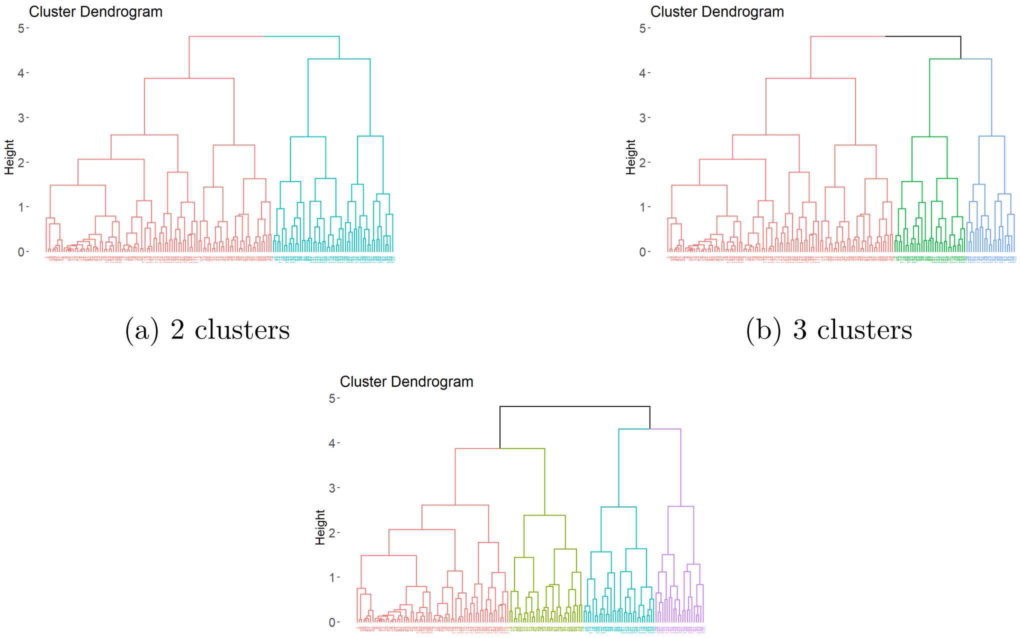
Figure 3.24: Silhouette index for AGNES clustering for average linkage.

3.2.4. DIANA

In the figure 3.25 we present the dendrograms for DIANA clustering. It looks similar to the AGNES method with an average linkage method. This observation can be confirmed by the Silhouette index presented in the figure 3.27.

Now, let's visualize our clusters. In the figure 3.26 we can see that the clusters are separated for 2 and 3 clusters, but they overlap for 2 clusters. In the figure 3.27 it can be observed, that the size of one cluster is larger than the others for all number of clusters. The average silhouette width is 0.39, 0.36, and 0.37 for 2, 3, and 4 clusters, respectively. For 2 clusters it works better.

Now, let's compare our clusters (for one where we use 2 clusters in the method) with the original classes. The rand index is equal to 0.67.

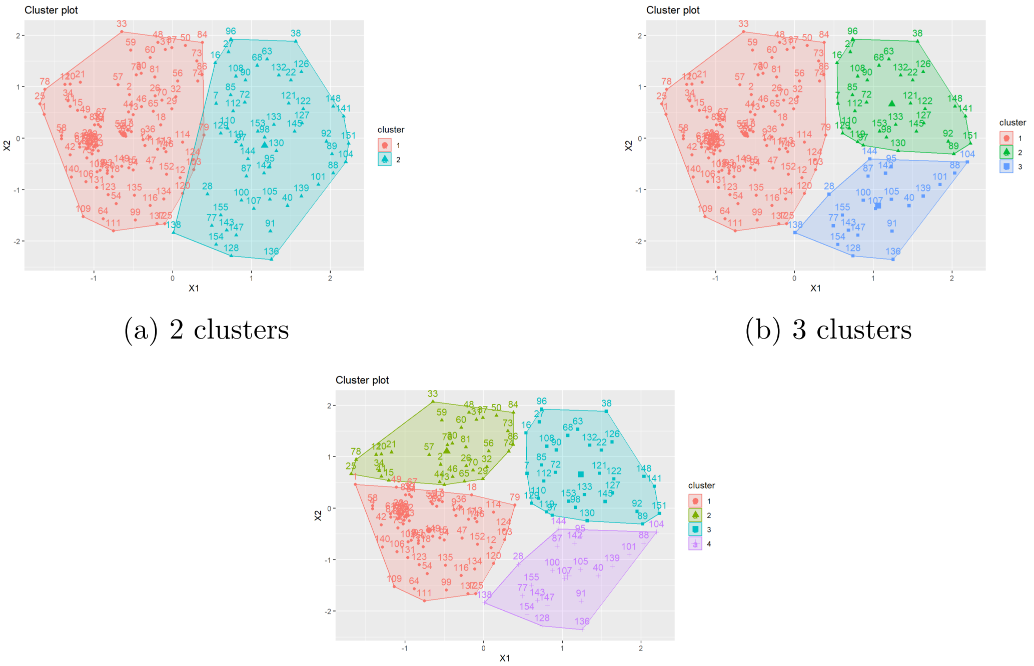


(a) 2 clusters

(b) 3 clusters

(c) 4 clusters

Figure 3.25: DIANA clustering.

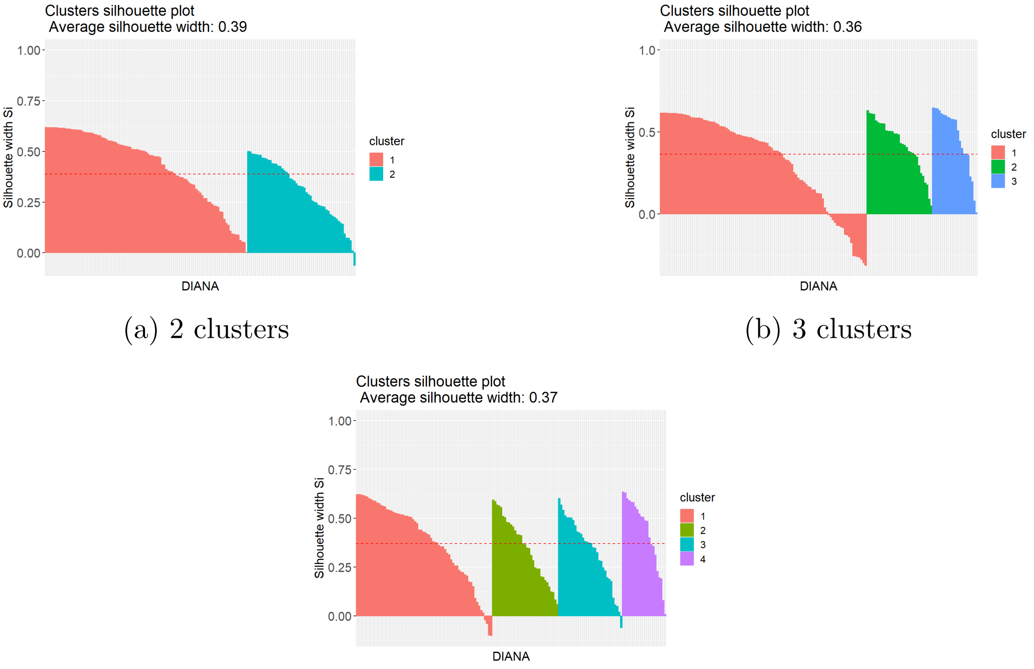


(a) 2 clusters

(b) 3 clusters

(c) 4 clusters

Figure 3.26: Visualization for DIANA method.



(a) 2 clusters

(b) 3 clusters

(c) 4 clusters

Figure 3.27: Silhouette index for DIANA clustering.

3.2.5. Fuzzy clustering

It is a partitioning cluster method, and we first have to select the number of clusters. Based on the figure 3.28, the optimal number of clusters for the Fuzzy clustering:

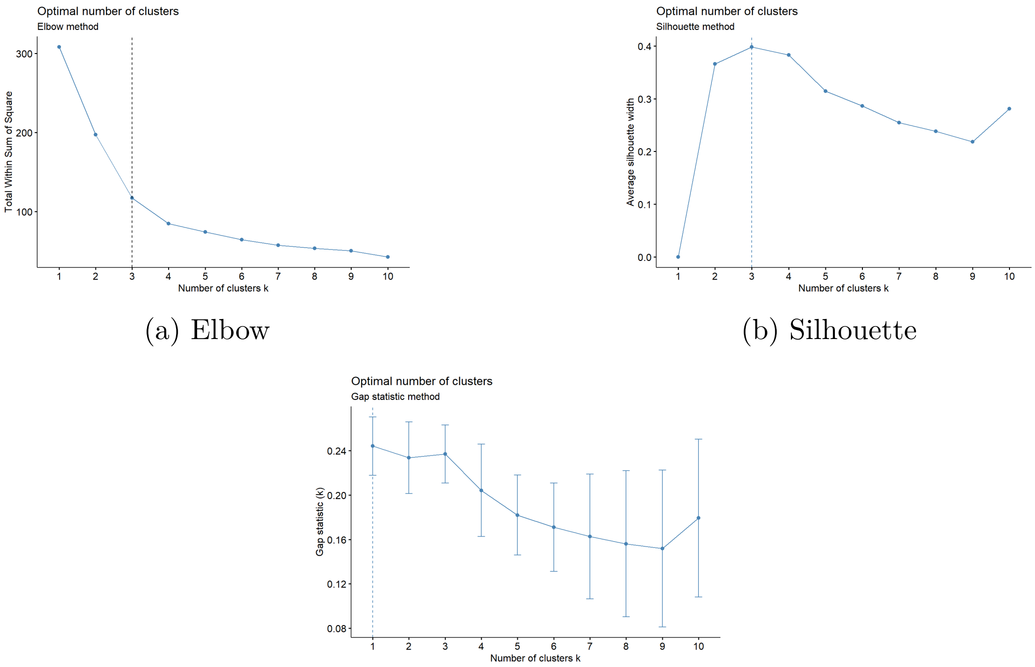
— Elbow method: it is hard to determine the optimal number of clusters because it is hard to say if it seems like the bend in the knee, but the possible choice is 3;

— Silhouette method: it says that the optimal number of clusters is 3;

— Gap statistic method: it says that the optimal number of clusters is 1, however, we also see that we can use 3 clusters.

Just looking at these statistics, we can use 2 or 3 clusters in Fuzzy analysis. Now, let's visualize our clusters. In the figure 3.29 we can see that the clusters are well separated for 3 clusters and overlap a bit for 2 clusters.

The last part of the analysis is to validate the clusters found. In the figure 3.30 we can see, that the size of the clusters is similar. The average silhouette width is 0.37, 0.4 for 2, 3 clusters, respectively. Now, let's compare our clusters (for one where we use 2 clusters in the method) with the original classes. The rand index is equal to 0.58.

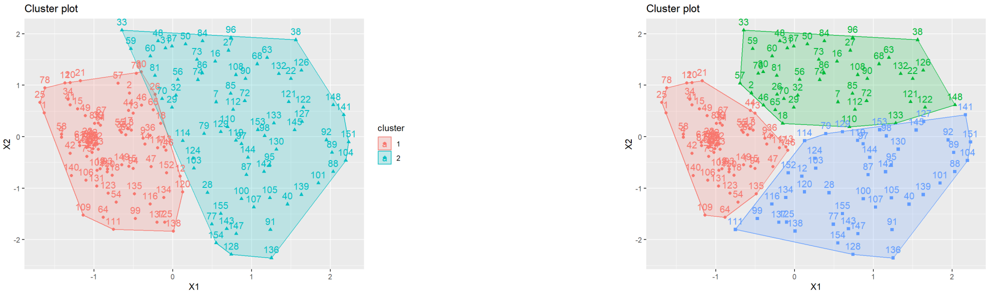


(a) Elbow

(b) Silhouette

(c) Gap statistic

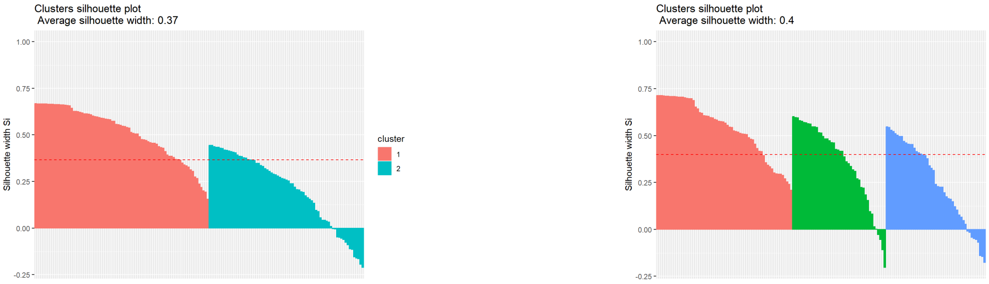
Figure 3.28: Optimal number of clusters for Fuzzy method.



(a) 2 clusters

(b) 3 clusters

Figure 3.29: Visualization for Fuzzy clustering.



(a) 2 clusters

(b) 3 clusters

Figure 3.30: Silhouette index for Fuzzy method.

3.2.6. Comparison

Now, let's compare all clustering methods. For almost all methods, we use 3 different numbers of clusters. In the table 3.8. We can see that the results are similar: clustering works badly for all methods. However, on the other side, the Silhouette index became better for all methods, excluding AGNES with average linkage. But the rand index often became worse. The worst models are AGNES with average linkage for 2 clusters (although the rand index is higher) and AGNES with complete linkage for 2 clusters. The best one is K-means for 4 clusters. To sum up, dimensionality reduction improved clustering for almost all models, but indexes suggested using more number of clusters. Moreover, the results are still disappointing.

	After MDS		Initial data	
Method	Silhouette	Rand	Silhouette	Rand
K-means for 2 clusters	0.38	0.58	0.30	0.66
K-means for 3 clusters	0.41	—	—	—
K-means for 4 clusters	0.42	—	—	—
PAM for 2 clusters	0.37	0.60	0.28	0.59
PAM for 3 clusters	0.41	—	0.25	—
PAM for 4 clusters	0.41	—	—	—
AGNES for complete 2 clusters	0.35	0.61	0.30	0.65
AGNES for complete 3 clusters	0.38	—	—	—
AGNES for complete 4 clusters	0.39	—	—	—
AGNES for average 2 clusters	0.28	0.70	0.32	0.70
AGNES for average 3 clusters	0.35	—	—	—
DIANA for 2 clusters	0.39	0.67	0.32	0.69
DIANA for 3 clusters	0.36	—	—	—
DIANA for 4 clusters	0.37	—	—	—
Fuzzy for 2 clusters	0.37	0.56	0.29	0.56
Fuzzy for 3 clusters	0.40	—	—	—

Table 3.8: Comparison of clustering methods.

4. Conclusions and further research suggestions

To conclude, it was difficult to work with medical data, because it consists of many missing values. We also had to address the class imbalance problem. These are common problems with this type of data. We also had a lot of categorical variables, so we had to choose our models and metrics carefully. During our analysis, we observed that using dimensionality reduction techniques may not always be a good choice for classification problems. To improve the medical diagnostic, it is better to analyze the features' importance and choose only the relevant attributes, such as Protime and Bilirubin. In unsupervised learning, dimensionality reduction gives us new possibilities for finding patterns in the data, but it is still not enough to capture all important relations.

As clustering methods, in general, perform poorly for our data, in the future we can consider some different methods, for example, more advanced clustering methods like OPTICS or Spectral Clustering. We can also try association analysis. Perhaps finding sequence or subgraph patterns will provide some interesting insights. Finally, we may consider different methods for dimensionality reduction, such as IsoMap (isometric feature mapping). It may perform better because it can handle complex, non-linear structures in data using geodesic distances.

In practice, our analysis can help to develop better diagnostic methods, discover which feature is more important, and which tests should be done first in hospitals to determine hepatitis in patients.