German Credit Data

Rafał Sokołowski

2023-01-21

Contents

1 Introduction

We move to the second part in the project, where we will discuss the *clusterization* and *dimensionality reduction*. Let's start with summarizing what we managed to do for now. The project stands from analysing **German credit data set**, in which our goal is to determine whether, the customer will pay the loan or he wouldn't do it.

As for now, we performed set of supervised methods for classification purpose. We discussed models as:

- LDA
- QDA
- K-Neighbors
- Random Forest
- TPOT - pipeline optimization using genetic algorithms

We conducted, that the best model was Random Forest with 80% AUC score along with 77% accuracy. It obtained even better result than TPOT optimization, which also found model from Random Forest family.

For the modeling purpose we performed EDA (exploratory data analysis) and basic data preparation, including:

- Dependency analysis, which resulted in dropping some of the features,
- Ordinal encoding - for categorical variables with defined order,
- One-hot encoding - for categorical variables without encoding,
- Passing through numerical values,
- Outlier detection using *Isolation forest*,
- Splitting data into train and test set.

We will use such obtained data set for our cluster analysis, to make the comparison between supervised and unsupervised methods more reliable. The main problem occurs from the scheme of our data. Most of features are categorical, which will result in violating some of the assumptions in methods that will use. We are aware that it will happen, but because such variables are the majority, we will go along with that and see what will happen.

Lastly, most of the columns are binary encoded flags (0 or 1). Comparing them with numerical values like, e.g. duration will generate huge bias during distance calculation. Because of that, we will standardize the whole data set before modeling. Additionally, we will fit the models using combined train and test set, but during the quality assessment, we will diverse between them.

2 Modeling

2.1 Methods used

2.1.1 K-Means

In K-Means algorithm we determine the membership of point to one of the K clusters. The algorithm:

- initialize K cluster centers $m_k = k = 1, ..., K$,
- Assign each point to cluster using the shortest distance,
- Calculate new cluster centers $m_k = \sum_{x_i \in C_k} x_i$,

where C_k are observations from cluster k, $k = 1, ..., K$.

2.1.2 Partition Around Medoids (PAM)

It's one of the generalizations of K-Means method, in which we also are seeking for cluster centers, but instead of calculating them, we will search for K representative objects (so-called *medoids*). The algorithm:

- From the set of n objects we select sequentially K representative objects, which are the initial centers of clusters,
- We minimize the sum of the distances of objects from corresponding medoids by replacing the curent medoids by other observations.

2.1.3 Aglomerative nestings (AGNES)

AGNES is a hierarchical clustering method. At the beginning, each object forms a separate cluster. In the next steps, the closest clusters are combined until one large cluster is created. The algorithm:

1. Each object forms a separate cluster,
2. We find the two closest clusters,
3. We join those clusters and replace them with one cluster,
4. repeat steps 2-3 until one large cluster is created.

We will analyse the algorithm using the complete linkage method.

2.1.4 Divisive Analysis (DIANA)

DIANA is a divisive algorithm, where in the first step all objects form one large cluster, which is then divided so as to obtain homogeneous groups.

2.1.5 Gower's dissimilarity measure

Most of the described algorithms can take as an input a *dissimilarity matrix* - the matrix which will determine the relation between observations and allow the proper grouping. Our data set consists of different types of features, including: ordinal, binary, continuous. From the lecture we know, that in general for such case we should use **Gower's dissimilarity measure**, defined as below:

$$d_{ij} = \sum_{k=1}^{p} w_k d_{ij}^k / \sum_{k=1}^{p} w_k, \text{ where}$$

- w_k 0 weight of attribute k,
- d_{ij}^k 0 distance (dissimilarity) calculated on the basis of k-th value.

2.2 Clustering - Internal indices quality assessment

We will start our analysis with determining the behavior of described clustering algorithms across different number of clusters, wherever we could, we used the dissimilarity matrix based on Gower's dissimilarity measure. For each case, we will compute the average silhouette score.

As we can see on Figure 1 the overall silhouette score is positive, so the clustering occurs, despite that the score itself isn't perhaps very high. What is relevant is that each algorithm obtained the highest score for $K = 2$ clusters, so we should expect that our observations can be divided into two groups. That conclusion

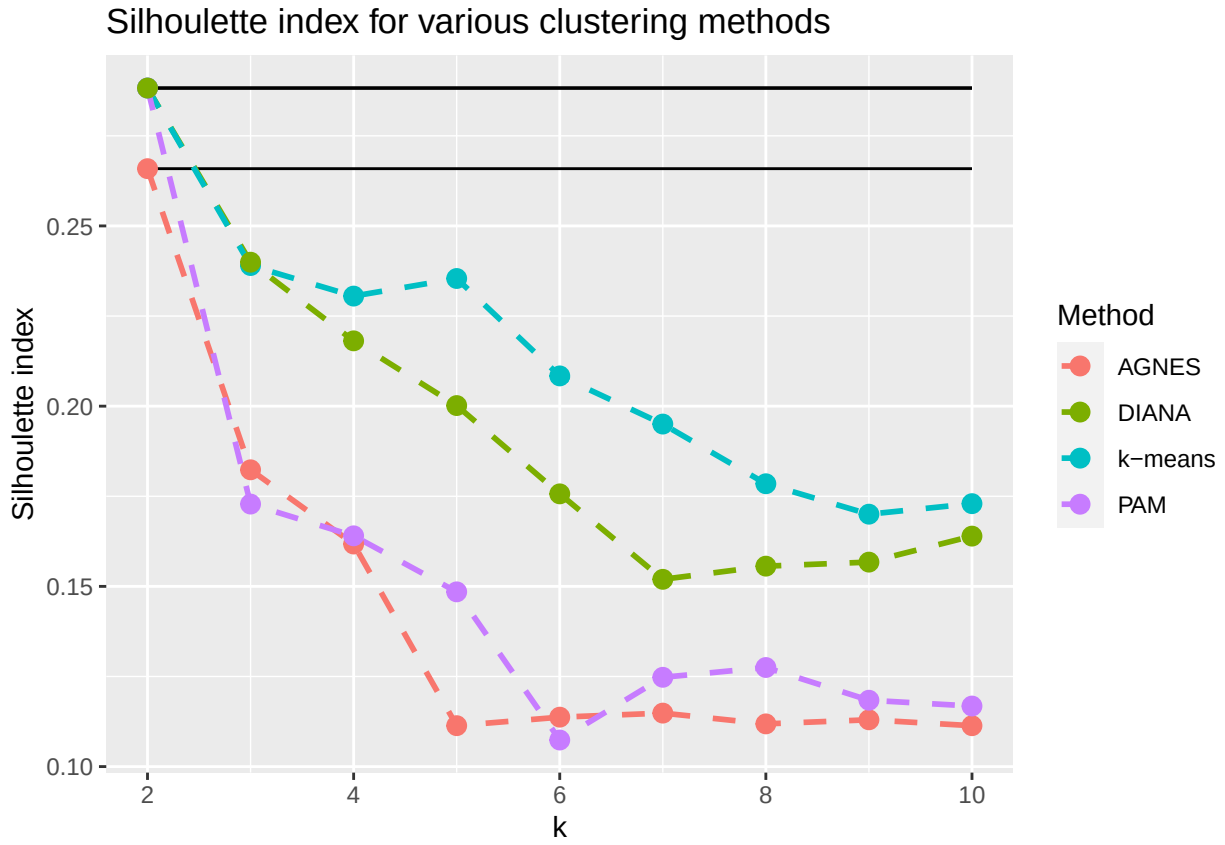

Figure 1: Comparison of silhouette score for different clustering methods against number of selected clusters. The solid lines represents the level of maximum silhouette score for each algorithm.

Table 1: Best clusterization algorithms with their corresponding numbers of clusters for specified internal indices.

	Score	Method	Clusters
Connectivity	0.500	kmeans	2
Dunn	0.615	kmeans	4
Silhouette	0.349	kmeans	3

pleases us, because our initial goal was a binary classification, so there were only two classes. Increasing the number of clusters will decrease the average silhouette, so it is better to keep low number of groups.

Apart from this analysis, we performed the cluster validation using *clValid* function in *R* for the same methods. The difference was, that we used Euclidean measure as a distance for clusters. This resulted in different values for silhouette score, but it can give us another view on how the methods could behave in such case. We inspect three measures:

- Connectivity,
- Dunn index,
- Average silhouette score.

The results of the optimal solutions can be seen in Table 1. As we can see, the K-Means dominates in all measures across the algorithms. Using the Euclidean distance, we improved the silhouette score, but it can be due to the fact, that some of the variables, despite the standardization, still out stands from other features. What is more, for different measures, the different number of cluster is selected. For more detailed analysis we can see Figure 2.

Let's start with connectivity, which for almost all algorithms is quite low, at least for small number of clusters. So the result for K-Means ($K = 2$) could be assigned for either AGNES and DIANA. For this case the PAM algorithm worked badly. The Dunn index also behaves similar for those 3 algorithms, and it keeps similar level for each $K = 2, 3, 4$, but of course the $K = 4$ identifies the most compact clusters. As we mentioned the silhouette was calculated with slightly other method, so we observe other result. Again K-Means, DIANA and AGNES behaves well and similar unlike PAM which can't manage it.

2.2.1 Silhouette score for different algorithms

We will analyse the behaviour of silhouette score for all observations and algorithms one by one. We will take under consideration the best case, for which the algorithms obtained the highest value, so for $K = 2$. Let's start with K-Means.

On the Figure 3 we can see the silhouette of each observation after using K-means method. The observations in cluster 2 are above average what is good, the bigger problems come from cluster 1 in which some of the observations don't fit. The algorithm keep the proportion of the data set. As we can remember, there were 30% of bad customers and 70% of good customers, those volumes may correspond to respectively cluster 2 and 1.

Let's take a look at the PAM results, which are presented on Figure 4. The results are very similar to those presented by K-Means, despite the assignment of cluster labels. It makes sense since both algorithms use similar methodology under the hood. Both algorithms move the centre of clusters to the best possible location.

On the Figure 5 the result is slightly different. The average silhouette score is slightly lower than previously and there are some observations which have negative silhouette score, so they are incorrectly assigned. But as we can see, the proportion of observations in cluster still holds.

Lastly, we can take a look at the Figure 6, where we can see results of DIANA. These are almost identical as for K-Means and PAM algorithms. The difference of the results with AGNES may occurred from specified linkage method. Perhaps by using some other method, the results for all 4 algorithms could be very similar.

2.3 Clustering - Quality Assessment

Presented algorithms are examples of unsupervised learning methods, so we don't know whether our results are good or not. Those methods tend to find some of interesting relationship within the data. Initially our task was classification and just for clustering purpose we discarded the labels. As we perform some kind of

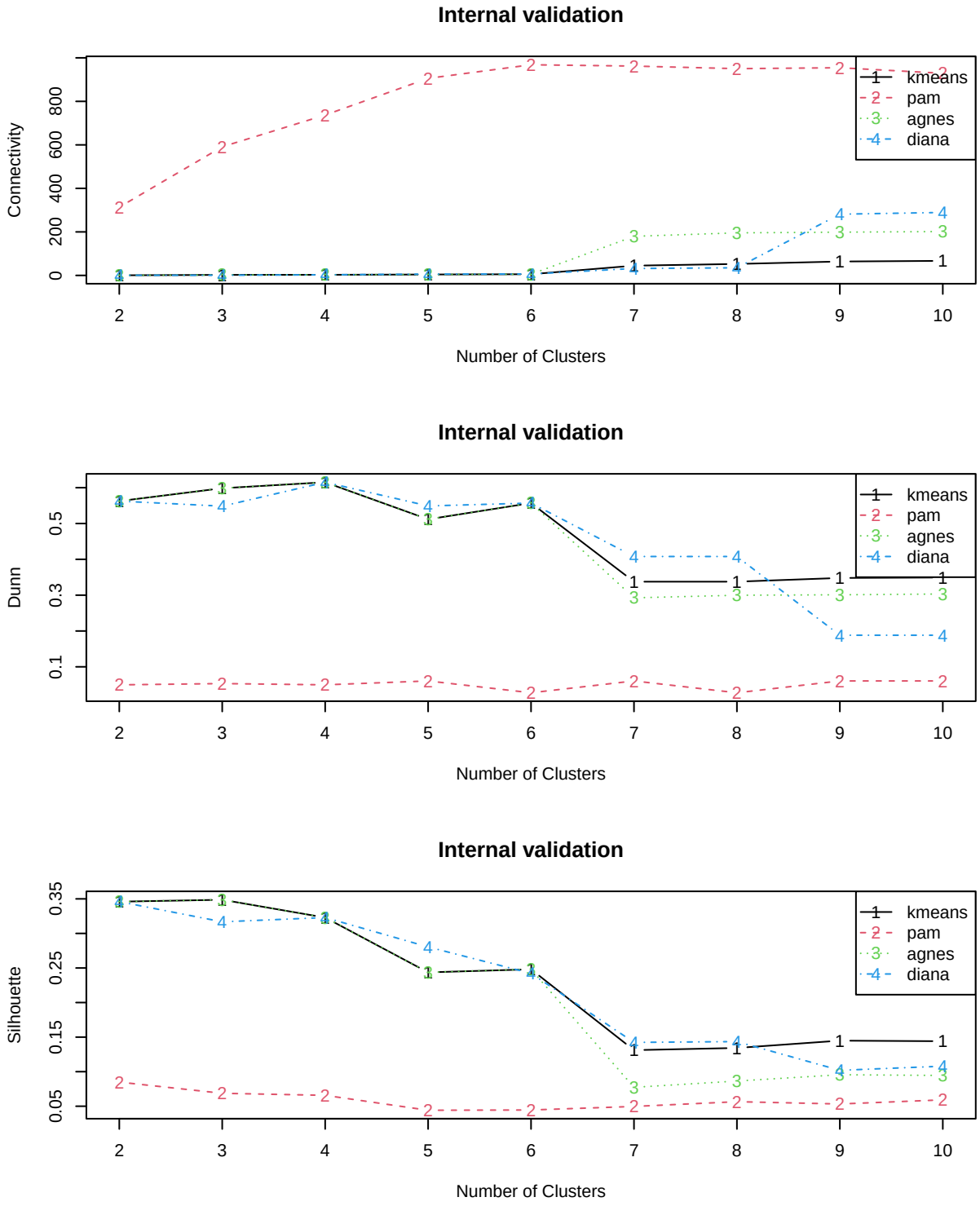

Figure 2: Different internal indices calculated accross set of algorithms with different numbers of clusters.

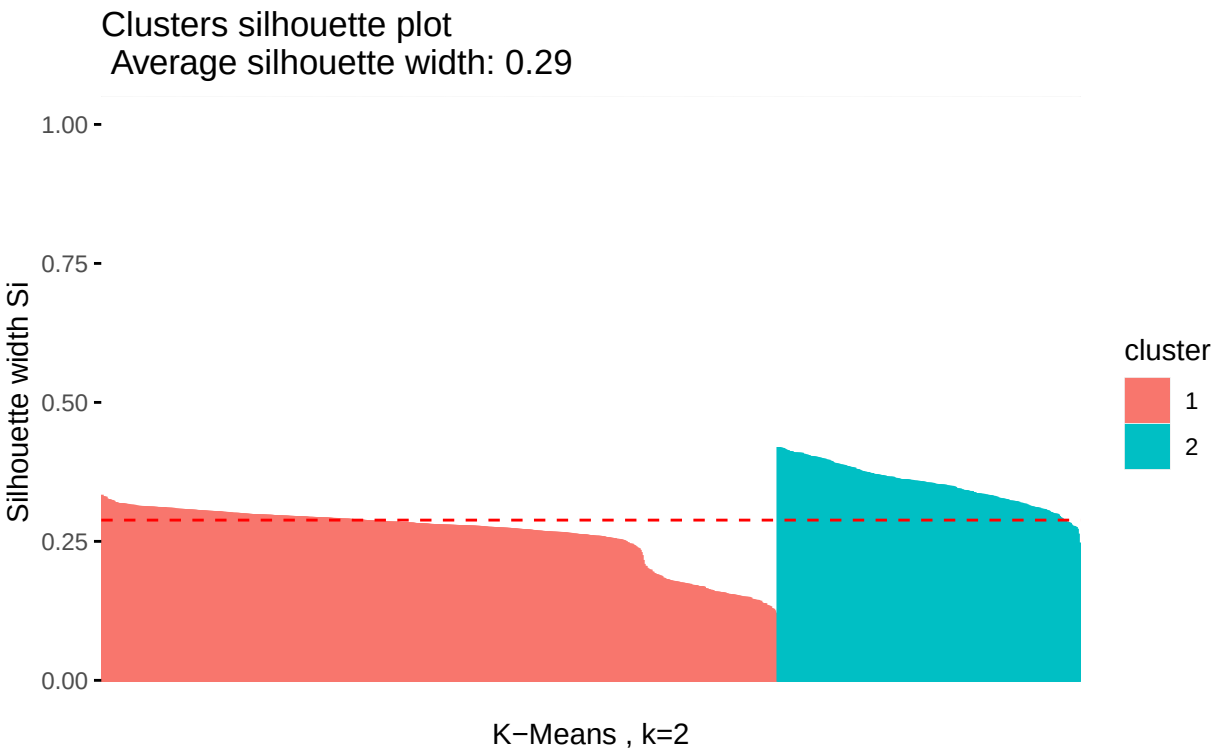

Figure 3: Cluster silhouette plot for k-means algorithm.

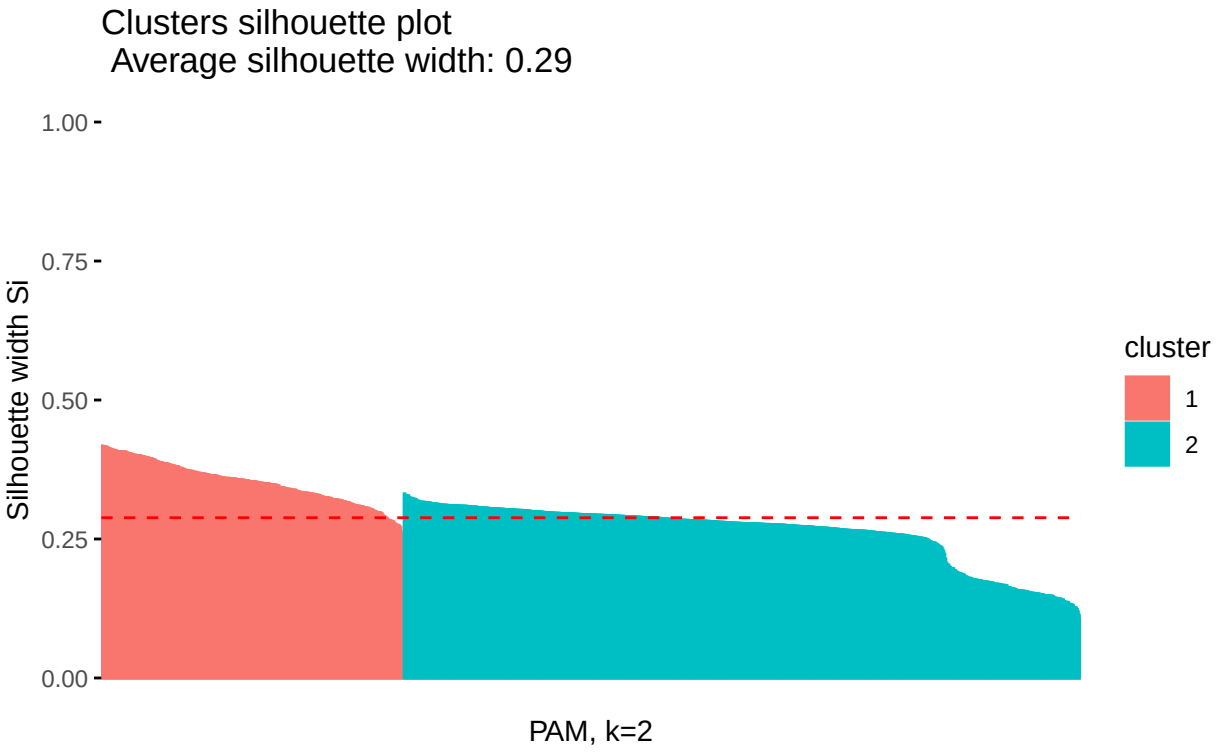

Figure 4: Cluster silhouette plot for PAM algorithm.

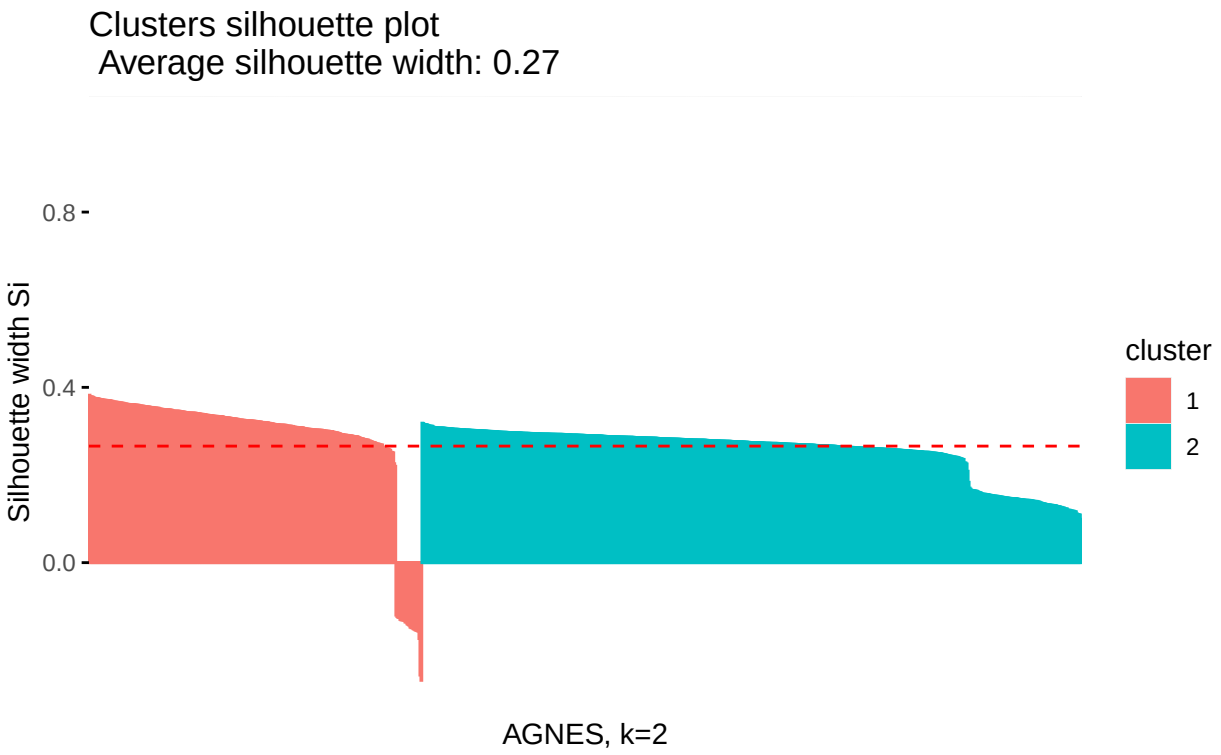

Figure 5: Cluster silhouette plot for AGNES algorithm.

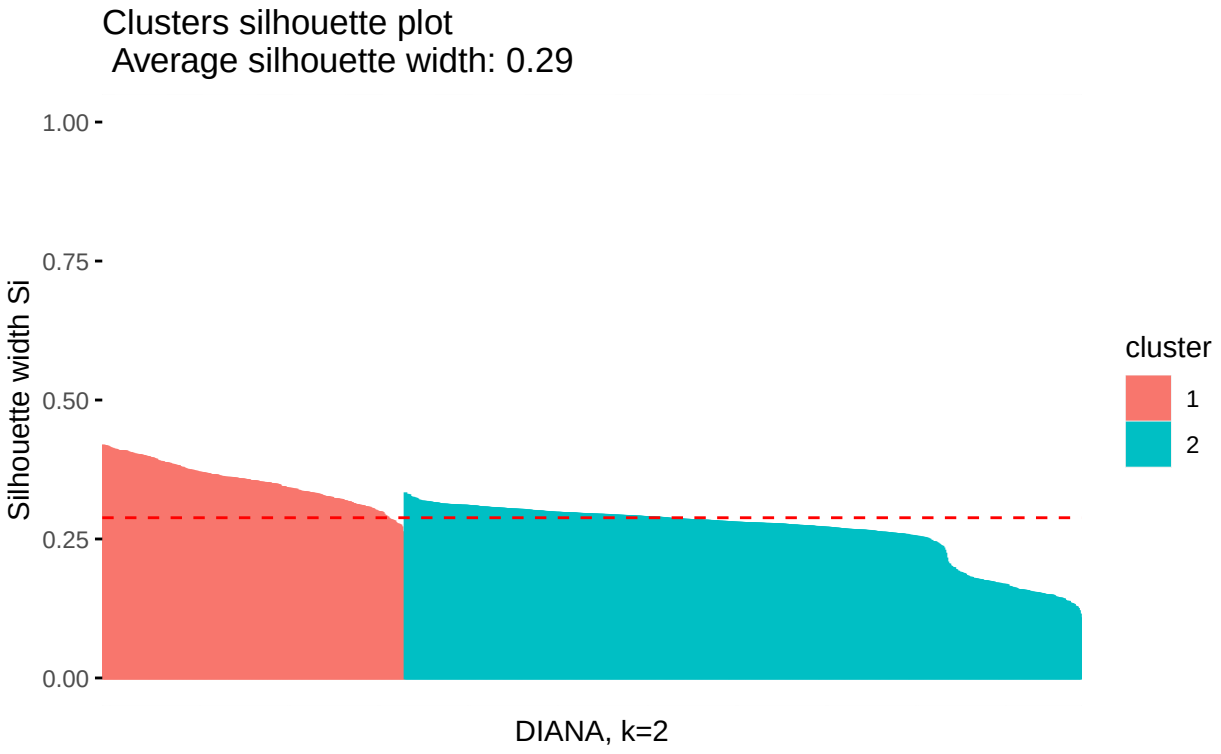

Figure 6: Cluster silhouette plot for DIANA algorithm.

Table 2: Results of all analysed algorithms in classifying bad customers.

	Accuracy.full	Accuracy.train	Accuracy.test	F1.test	Recall.test	Precision.test
K-Means	0.607	0.601	0.606	0.371	0.39	0.354
PAM	0.607	0.601	0.606	0.371	0.39	0.354
AGNES	0.589	0.592	0.581	0.357	0.39	0.329
DIANA	0.607	0.601	0.606	0.371	0.39	0.354
LDA	NA	0.760	0.740	0.480	0.41	0.600
QDA	NA	0.770	0.770	0.590	0.56	0.630
K-Neighbors	NA	0.740	0.720	0.430	0.36	0.550
Random Forest	NA	0.960	0.770	0.630	0.66	0.600
TPOT	NA	0.930	0.740	0.400	0.29	0.630

analysis, we can move back and check the quality of algorithms. In the Table 2 we can see a set of metrics calculated for clustering algorithms along with the results from previous classification.

We trained our cluster algorithms using the whole data set, but we can select the observations which initially were assigned for the test set. We know how the results looked for the supervised methods and now we can compare them for clustering methods. In this place we should mention, that it is difficult to compare the real class labels with the labels assigned by clusters. Such assignment is often random and is unreliable. Having that, as a preliminary step, we performed optimal class assignment. As a sanity check for that, we can look at the column *Accuracy.full* which was calculated only for cluster algorithms (because supervised methods weren't evaluated on the whole data set). We can be sure that it was done properly if the value is higher than 0.5, because if it's not, then the labels should be assigned conversely.

We can see, that the clustering algorithms obtained very similar results on the level of 60% accuracy, and only the AGNES performs slightly worse. If we look at the metrics on the test set, then we can see that in fact, the algorithms perform poorly for such task. The classifying bad customers is very important, and analysed methods cannot manage that. All scores are below 40%. What is natural, the classification methods performs better than the clustering methods. It makes sense, because they used additional information about the customers during the learning process, which was the labels. The conclusion is that, perhaps there are no clear rules which could determine if customer is good or bad judging by such information.

3 Dimensionality reduction

We will use *dimensionality reduction* methods to transform our initial feature space into space in lower dimension. Such operation may result in removing redundant variables from the data set and clarify some of the hidden relationship between it and the label. Lots of transformations can result in handling the problem of ordinal and binary variables, turning them into continuous one. It can lead to in better behaviour of our clustering methods, as well as the classification methods.

3.1 Used methods

3.1.1 Principal Component Analysis (PCA)

PCA is an algorithm which transforms original feature space into another using some sort of linear combination. In other words, we want to transform our variables $X_1, X_2, ..., X_p$ into so-called *principal components* $PC_1, PC_2, ..., PC_p$, where $PC_i, i = 1, ..., p$ are sorted and the PC_1 has the highest variability. Setting threshold of overall data variability that we want to keep, we can reduce the space dimensionality by discarding the least "informative" variables.

3.1.2 Multidimensional Scaling (MDS)

As PCA, MDS tends to reduce the data dimensionality, but this time it tries to preserve original distances between objects. Regarding the fact, our variables are of different type, we will use proper dissimilarity matrix, more specifically using Gower's dissimilarity measure.

3.2 Projecting with PCA

Let's start by transforming our data set with PCA algorithm. Using the specification of the method, we can analyse the *explained variance ratio*, so the normalized variability of each principal component. Furthermore, we can inspect, the *cumulative explained variance ratio* which will show us how many principal components should we take in order to keep some level of information in our data set.

Both information were visualized in Figure 7. As we expect, the explained variance will decrease as the number of principal component increase. For features above PC_{30} are these whose variability is even 0, so definitely we will discard them. We can see that the PC_1 holds about 9% of the whole data set variability which, regard to number of features, is quite high. In later analysis, we will constrain ourselves to keep the 90% data variability, as it is often done in practice.

For now let's take advantage of other property which dimensionality reduction allows us, so let's visualize the data in 2D space using PC_1 and PC_2. The result is presented on Figure 8. As we can see, observations are grouped into 2 separate clusters. It may indicate the fact, that lots of our initial features were one hot encoded, which lead to such shifted shape. Despite, the fact that we have two independent groups, if we mask them using true labels, we can see that they are completely mixed. There is no clear relation between the PC_1, PC_2 and the customer attitude. At this moment, we can expect, that the performance of the algorithms perhaps wouldn't be better than previously.

For further analysis, we can train our clustering algorithms on, e.g. 10 first principal components and check, how they will perform. Following that, we will visualize its predictions on the lower 2D space. Such information can give us hint if despite no significant relations with two first components, we can obtain some better results using more features. The results are presented on Figure 9. As we could expect, the algorithms focused mainly on separating those two clearly visible groups, although it won't improve the classification of customer's attitude. So in most cases, one group is marked with triangles and the other is marked with circles. Interesting part happens in AGNES algorithm where in spite of separating observations by those two groups, it tries to slice both of them using one skew line.

3.2.1 Selecting valid number of principal components

We will perform cluster analysis with valid number of principal components. We selected the $n + 1$ number of components, where n was the number of principal component, which sum of the explained variance ratio of the next component and all previous would result in exceeding 90%. So in our analysis $n = 25$. On the Figure 10 we can see the average silhouette score for all algorithms trained on constrained data set by different number of clusters. In most cases the score dropped . Otherwise, the AGNES algorithm, which previously performed the worst, now is the best. It's highest average silhouette is around $K = 3$ number of clusters, but because it's almost identical as for $K = 2$ then for simplicity we will stick to $K = 2$.

An interesting observation occurs for K-Means which has the same level of average silhouette score for all considered number of clusters. So, it will always try to separate the observations identically.

In the Table 3 we can see the highest average silhouette score for all algorithms, As we can remember before PCA, all numbers were around 0.3, but now we observe an outstanding improvement for AGNES algorithm.

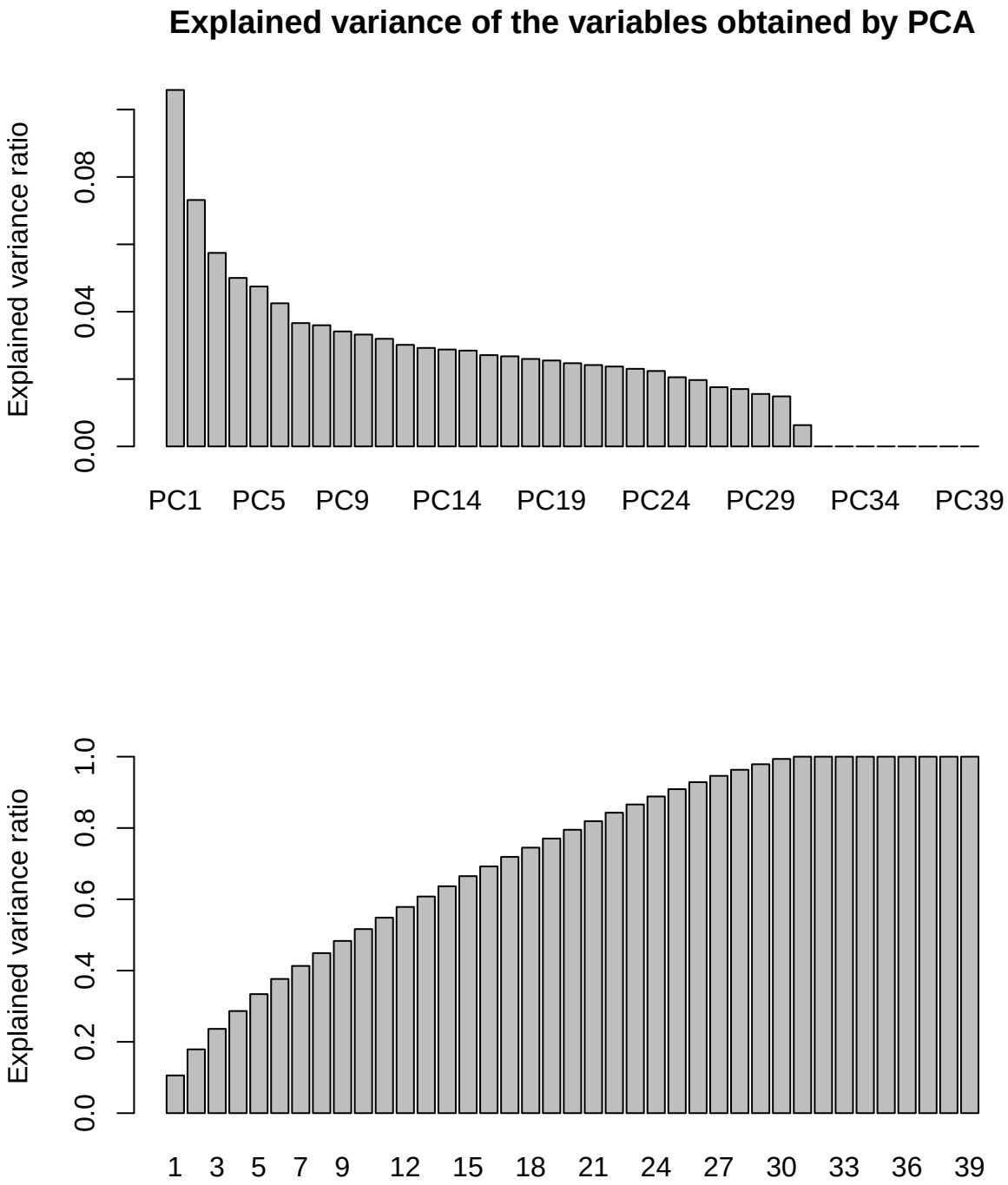

Figure 7: Explained variance ratio and cumulative explaned variance after using PCA on our data set.

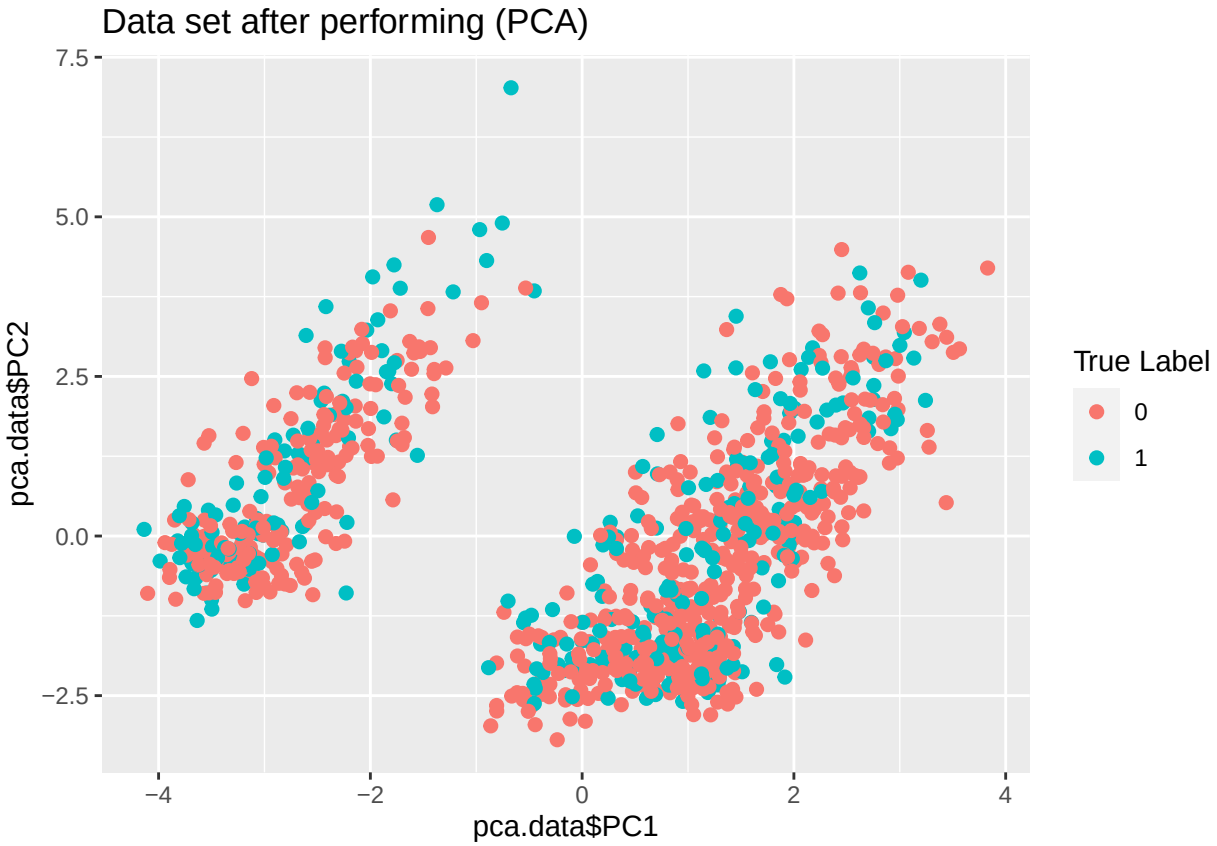

Figure 8: Visualization of data set after PCA in 2D. We plot PC1 versus PC2 and just for reference we mark observations using the true customer's label.

Table 3: Silhouette score for clusterization algorithms after performing PCA.

	x
AGNES	0.358
DIANA	0.092
k-means	0.113
PAM	0.122

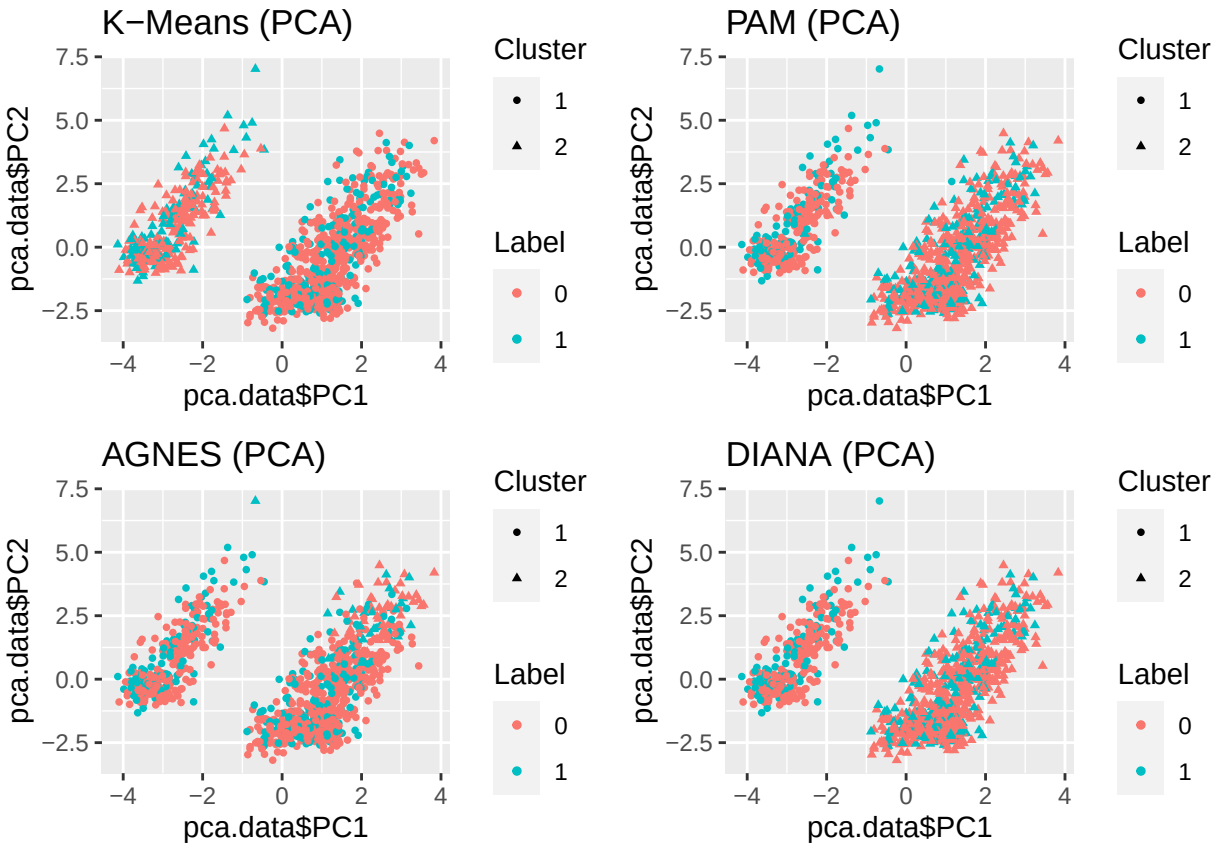

Figure 9: Visualization of data set after PCA in 2D. We plot PC1 versus PC2 and just for reference we mark observations using the results of considered clusterization algorithms vs true labels.

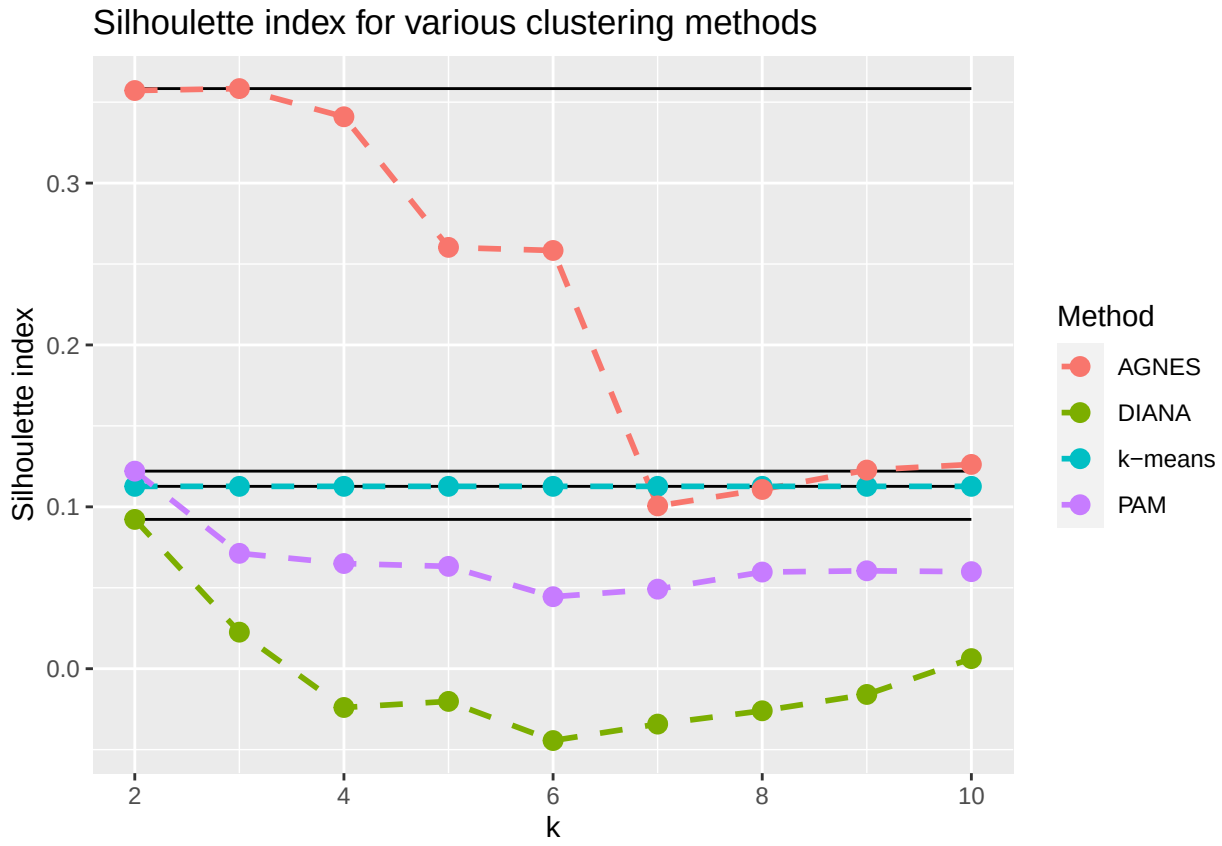

Figure 10: Comparison of silhouette score for different clustering methods after using PCA against selected number of clusters. The solid lines represents the level of maximum silhouette score for each algorithm.

3.3 Projecting with MDS

We will now continue the same procedure for MDS algorithm. Using the specification of method, we will perform STRESS vs. dimension analysis presented during the lecture. It will enable us to select proper number of dimensions used for the algorithm.

Figure 11: Visualization of data set after MDS in 2D. We plot X1 versus X2 and just for reference we mark observations using the true customer's label.

Let's begin with visualizing our transformation in 2D. This time we specify number of dimensions used for the algorithm to 2, and visualize the results on Figure 11. This time we don't obtained two separate groups, but four, regarding that two of them consists of two subgroups. One more time, despite the fact that we see separate groups, they don't indicate any significant relations between the customer's attitude. Let's take some number of features, e.g. once more 10 and train clustering algorithms on them. Then visualize obtained results in 2D.

From the Figure 12 we can see, that in most cases, the clusterization results also tends to separate visualized groups. This time something unexpected happens for PAM algorithm, where we can observe more diversity among the clusters. This time the number of dimensions were done arbitrary. Let's make it more consciously using STRESS vs dimension plot.

As we can observe on @ref{fig:stress-mds}, from some point the level of STRESS starts to increase. Our goal is to minimize that measure, so we will choose the resulted number of dimensions to be 7, as for this case we obtain the minimal value of STRESS.

On the Figure 14 we can see the average silhouette score for all algorithms trained on constrained data set from MDS by different number of clusters. This time the behaviour is more similar to the situation before dimensionality reduction. The optimal number of cluster is 2 for all algorithms and they oscillate mostly

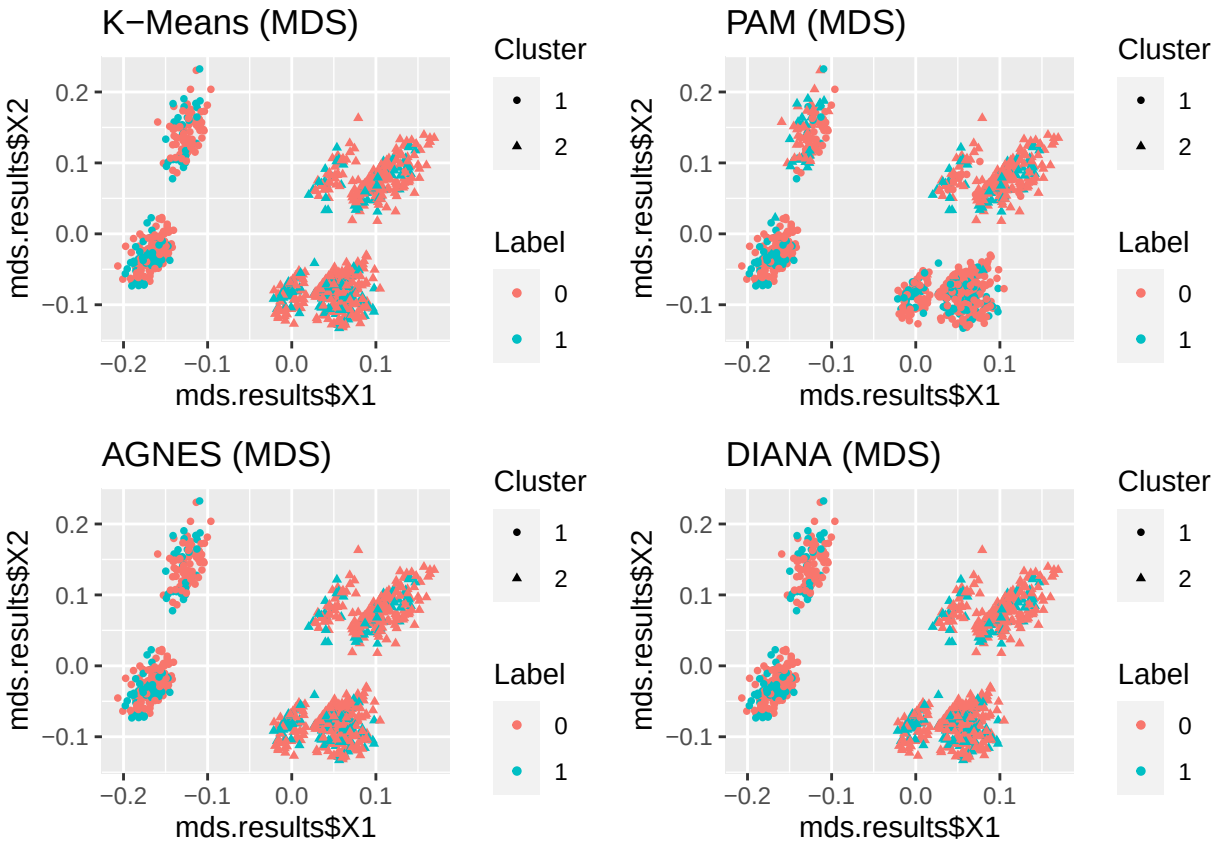

Figure 12: Visualization of data set after MDS in 2D. We plot X1 versus X2 and just for reference we mark observations using the results of considered clusterization algorithms vs true labels.

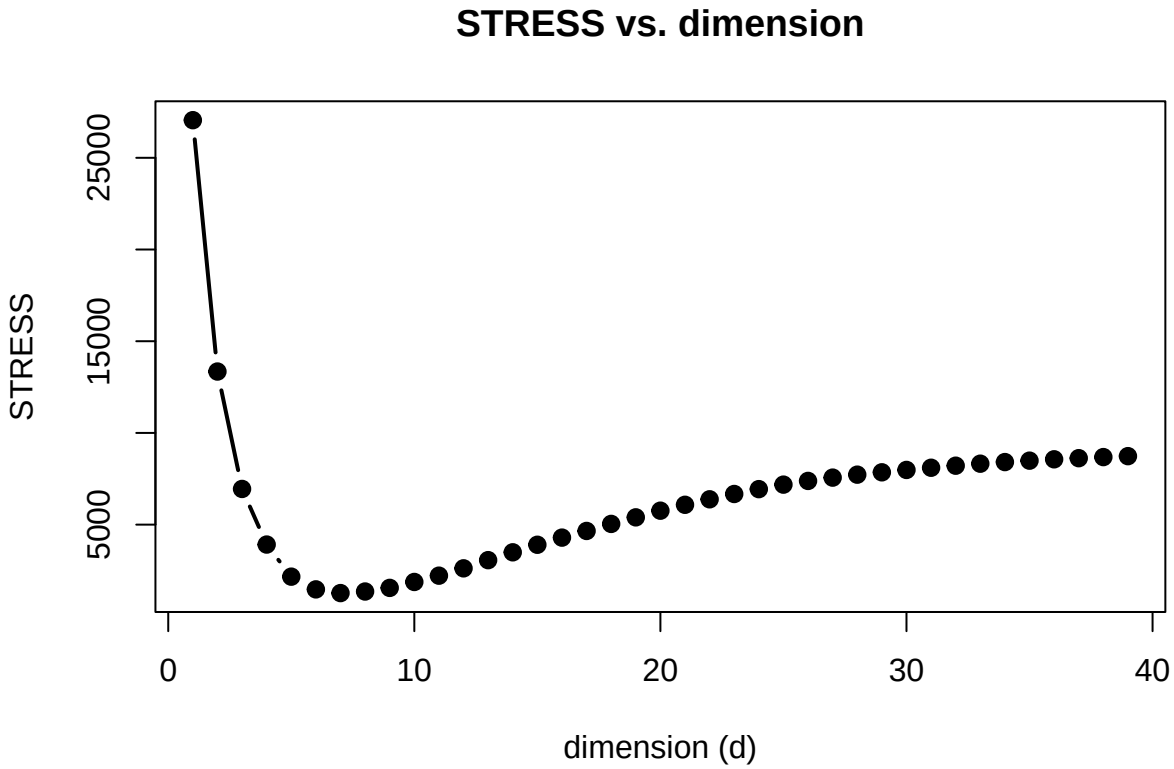

Figure 13: STRESS vs dimension. We try to minimize the level of STRESS by selecting proper number of features.

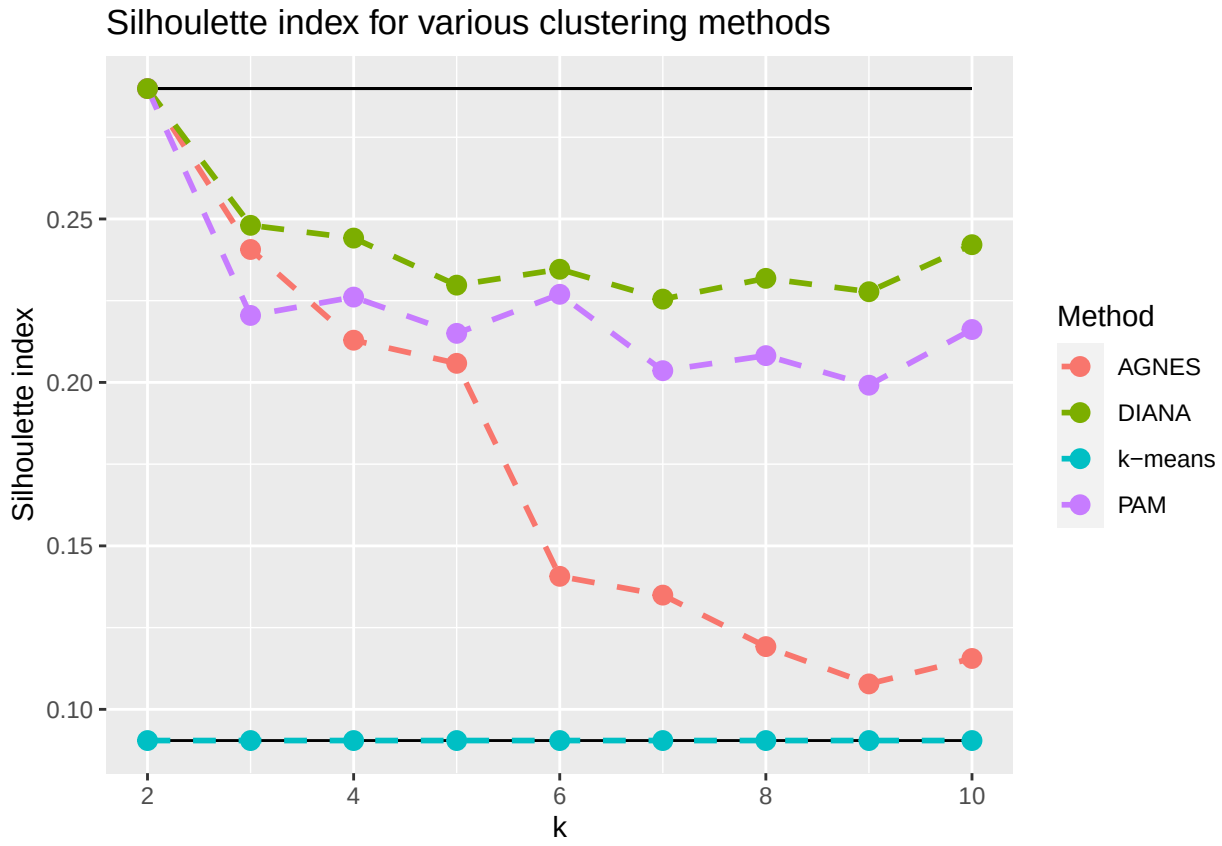

Figure 14: Comparison of silhouette score for different clustering methods after using MDS against selected number of clusters. The solid lines represents the level of maximum silhouette score for each algorithm.

Table 4: Silhouette score for clusterization algorithms after performing MDS.

	x
AGNES	0.29
DIANA	0.29
k-means	0.09
PAM	0.29

around 0.3. The difference happens for K-means which one more time presents constant silhouette average score for different number of clusters, but its value is the lowest from all cases.

The detailed values of highest average silhouette score can be found in Table 4. As we discussed, nearly all algorithms have this value around 0.29, but for K-means it is very low on the level of 0.056.

3.4 Results

We performed analysis of clusterization for PCA and MDS. We will move to quality assessments of used methods similar to the previous one. All results were presented in @ref{tab:all-results}, where to previously assessed algorithms, we add evaluations after dimensionality reduction.

The analysis were also performed for supervised methods, but in this case we omit the advanced hyperparameter tuning and just take the output of PCA and MDS, split the data into train and test set and train the models with their hyperparameters obtained during the previous analysis.

The results for PAM and K-Means despite different befavior during assessment of dimensionality reduction don't change. For AGNES we can obtain slight improvement after using MDS and relevant improvement using PCA which agrees with our previous conclusions. What is more interesting is that, the DIANA using PCA obtained identical results, despite previous differences. Regarding that, the MDS doesn't change anything for DIANA algorithm.

Following that, we may conclude that the linear transformation of data set was needed for hierarchical clustering to obtain such result. In case of AGNES the slight improvement with MDS may come from the chosen linkage method for the algorithm. Interesting is that for partitioning methods like K-Means and PAM there are no differences in results, but maybe it's the case of the size of data.

For the supervised methods, the results are looking slightly different. For LDA, we can see an improvement using PCA. It may come from the previously violated assumption, that LDA needs the continuous values, which this time it received. Also, the fact that it is a linear model and the PCA performs linear combination on data could have impact on such behavior. There is no improvement after using MDS. The situation differs for QDA.

Last time we also violated assumptions of the algorithm, but the dimensionality reduction doesn't improve the accuracy. Subsequently, the precision/recall for the MDS is tragic, but we obtain significant improvement of those metrics for the PCA case. The decrease of accuracy had to be in cost of misclassifying the good customers, but our interest is in proper classification of bad ones. Regarding that, the algorithm received great result.

The same behavior as for the LDA occurred for the K-Neighbors algorithm, but the improvement isn't so big as for LDA.

For the random forest, the dimensionality reduction also doesn't improve the results. It looks that some important properties of features, from the point of view of this algorithm, disappeared.

The clustering algorithms still performs worse than supervised methods, which is an natural observation. What is relevant, that despite the whole dimensionality reduction analysis the LDA, which obtained because of it the best accuracy across all considered algorithms along with PCA and MDA, still don't beat the initially trained random forest. The other case happens for QDA along with PCA. It indeed has worse

Table 5: Results of all analysed algorithms in classifying bad customers.

	Accuracy.full	Accuracy.train	Accuracy.test	F1.test	Recall.test	Precision.test
K-Means	0.607	0.601	0.606	0.371	0.390	0.354
K-Means (PCA)	0.607	0.601	0.606	0.371	0.390	0.354
K-Means (MDS)	0.607	0.601	0.606	0.371	0.390	0.354
PAM	0.607	0.601	0.606	0.371	0.390	0.354
PAM (PCA)	0.607	0.601	0.601	0.358	0.373	0.344
PAM (MDS)	0.607	0.601	0.606	0.371	0.390	0.354
AGNES	0.589	0.592	0.581	0.357	0.390	0.329
AGNES (PCA)	0.699	0.696	0.697	0.032	0.017	0.333
AGNES (MDS)	0.607	0.601	0.606	0.371	0.390	0.354
DIANA	0.607	0.601	0.606	0.371	0.390	0.354
DIANA (PCA)	0.699	0.696	0.697	0.032	0.017	0.333
DIANA (MDS)	0.607	0.601	0.606	0.371	0.390	0.354
LDA	NA	0.760	0.740	0.480	0.410	0.600
LDA (PCA)	NA	0.760	0.770	0.570	0.510	0.640
LDA (MDS)	NA	0.710	0.710	0.440	0.370	0.520
QDA	NA	0.770	0.770	0.590	0.560	0.630
QDA (PCA)	NA	0.760	0.670	0.640	0.720	0.580
QDA (MDS)	NA	0.700	0.700	0.000	0.000	0.000
K-Neighbors	NA	0.740	0.720	0.430	0.360	0.550
KNN (PCA)	NA	0.770	0.740	0.450	0.360	0.600
KNN (MDS)	NA	0.750	0.680	0.290	0.220	0.420
Random Forest	NA	0.960	0.770	0.630	0.660	0.600
RF (PCA)	NA	0.980	0.720	0.460	0.410	0.530
RF (MDS)	NA	0.930	0.660	0.410	0.390	0.430
TPOT	NA	0.930	0.740	0.400	0.290	0.630

accuracy, but it outperforms random forest in case of precision and recall, which is far more relevant to us. So if we would like to focus on classifyng bad customers, we should use QDA preceded by PCA.

4 Conclusion

We tried to perform cluster analysis with comparison to supervised methods. Then we perform similar analysis after using dimensionality reduction. Regarding that, we could reevaluate some of our conclusions from previous part of project. Let's recall that our major goal is to classify if the creditors will be good or bad customers, with more weight put on classifying bad customers. We used data about characteristics of customers from both groups to train different algorithms. The task from the begining was non-trivial and despite using advanced methods, the results on the test set were below practical values of precision and recall. Our major objective was to be more precise in misclassifying bad than good customers and using the bad customers as a positive class, the recall was at most 66% from the tuned random forest.

The analysis could have reassured us that the clusterization algorithms performs worse than supervised methods, even after using dimensionality reduction. Nonetheless, those exercise could show us that there are no clear relations between features about customer and their attitude to paying the loan. The algorithms indeed, found that there are two groups in data, but they completely couldn't translate them into classification groups.

The analysis of dimensionality reduction with addition of the supervised methods resulted in finding that the QDA along with PCA may outperform random forest in case of classifying bad customers. It is worth outline, that there could extend the analysis of dimensionality reduction with classification methods using some kind of hyperparameter tuning. So we could allow models to give their best in the evaluation. In my opinion after such operations random forest should one more time outperform the QDA algorithm.

In conclusion, despite the methods and algorithms used in the project there may be some other methods which could increase the performance, like some from the field of deep learning. The model's user should keep in mind that its prediction isn't ideal, and the presence of bad customers may occur. The analysis could be extended on the financial reports concerning how this classification results could influence the solvency of the bank.