Data Mining

Course project

Marta Kawalko (229955), Zuzanna Materny (229932)

# Application of data mining techniques on *Breast Cancer Wisconsin* data set

part II

# Contents

# 1 Introduction

The project goal is to use remaining data mining methods to perform complete analysis of selected data. A great number of techniques have been applied to our analysis in the previous part 2. In this project we will mostly focus on cluster analysis with quality assessment as well as the chosen dimension reduction methods in connection with clustering and classification. At the same time we will deepen our knowledge of the real breast cancer problem.

As mentioned before in 2 breast cancer is one of the most common cancers women are facing. It is a serious problem that cannot be neglected. The analysis is carried out on the breast cancer data obtained from the University of Wisconsin Hospitals ([1]). The data was collected in the years 1989-1991 and contains the following information:

| | Attribute | Domain |
|---|---|---|
| 1. | Sample code number | id number |
| 2. | Clump Thickness | 1 - 10 |
| 3. | Uniformity of Cell Size | 1 - 10 |
| 4. | Uniformity of Cell Shape | 1 - 10 |
| 5. | Marginal Adhesion | 1 - 10 |
| 6. | Single Epithelial Cell Size | 1 - 10 |
| 7. | Bare Nuclei | 1 - 10 |
| 8. | Bland Chromatin | 1 - 10 |
| 9. | Normal Nucleoli | 1 - 10 |
| 10. | Mitoses | 1 - 10 |
| 11. | Class: | (2 for benign, 4 for malignant) |

# 2 Reminder of the conclusion drawn in part I

## 2.1 Reasearch already conducted

After familiarizing with the data set, we focused on both missing values and identification and interpretation of outliers. Then multiple feature selection methods were applied. We have examined Boruta algorithm, random forest methods with conditional inference trees as well as Breiman's algorithm. Stepwise feature selection was also used to analyse the best subset of the data set. From that we have learnt that there exist less relevant features which can be omitted without the negative impact on the analysis results. In the further chapters of the project the class imbalance problem was considered. For better classification results many various methods were inspected. Linear regression model was the starting point. Then the $k$ nearest neighbours algorithm was applied along with Linear Discriminant Analysis, which was followed by Quadratic Discriminant Analysis. Logistic regression came along right after. At the very end we have considered single classification tree and whole random forests.

Most of the methods we discussed were applied to numerical data. We kept in mind that the values in our data frame were actually categorical, but that worked very well. Only classification tree and random forest worked directly on data of type factor.

Carrying out cross validation assured us about their good performance. We decided not to repeat feature selection in each iteration in favour of generalizing the importance for all measures and making model independent of training/testing subset division. It turned out that three variables are sufficient to predict the class of tumor with good accuracy.

## 2.2 Final conclusions

To draw proper conclusions, we should consider the purposes of the whole analysis and classification. If it is supposed to be used for only statistical purposes, having information about Bare Nuclei, Clump Thickness and Uniformity of Cell Size will give us satisfactory results. When it comes to patient's diagnosis, we should be as accurate as possible, so rather use full models or models excluding Mitoses and Marginal Adhesion,

since our analysis proved their negligible importance. Giving less attention to type I and type II errors might result with strongly confusing diagnoses, which might have fatal consequences.

# 3 Cluster analysis

On of the examples of unsupervised learning is cluster analysis. We wish to detect the internal data structure and create the proper number of clusters – separated and homogeneous groups. We have to make sure that the distance between the similar objects in one cluster along with the dissimilarity measure between objects is determined accurately. We try to figure out how many clusters should be created and which objects belong to which group. Such work can allows us to formulate new hypothesis after reveling characteristics that were hidden before. Dimension reduction can also be the next step after clustering. We may also expect obtaining different results and solutions after using various clustering algorithms. Keeping in mind that such approach is unstable, we will handle clustering carefully and deliberately. The interpretation of the results may be a hard nut to crack.

There is a popular split between clustering: soft and hard clustering. Soft clustering allows an object to belong to more than one cluster with some likelihood value or probability. On the contrary, hard clustering requires that the object belongs to one and only cluster. In our analysis we will mostly focus on hard clustering.

There are plenty of different algorithms regarding clustering. As it is the subjective task, there can be few correct clustering algorithms. We will experimentally decide which algorithms along with specified parameters are the best for our data set.

Clustering is also a tool that helps us understand and explore the data, but we have to be careful because single clustering might be misleading and unreliable.

## 3.1 Partitioning methods

Partitioning methods help us find the split into $K$ groups so that the differences between the objects in one group are optimally small. We just have to specify the number of groups $K$ and the criterion – dissimilarity or distance measure.

### 3.1.1 $k$-means method

One of the most commonly used algorithm involving the division into specified number of clusters is called $k$-means. In this algorithm objects belong to the cluster with the nearest mean. The method uses squared Euclidean distance to optimize the variances within each cluster. The sum of such distances is the total within-cluster variation:

$$V(C_k) = \sum_{x_i \in C_k} (x_i - \mu_k)^2,$$

where $x_i$ is an object that belongs to cluster $C_k$ and $\mu_k$ is the mean value for cluster $C_k$.
For our data set the intuitive number of clusters is 2 as we have 2 types of cancer: benign and malignant. If we do not know the proper number of clusters, we usually check the results and decide. We can also consult a doctor as we use the medical data set.

Let us start from checking the results for different number of clusters. We will consider 12 clusters as our maximum. We have to keep in mind how the number of partitions grows along with the number of clusters. The first step is to remove class labels as it is unsupervised learning. We will also create a table working like a confusion matrix and telling us how many observations are in "benign" cluster and actually representing benign cancer and how many observations belong to the other cluster and the same thing for malignant cancer. Based on the table the ratio of correctly matched pairs is calculated.

```
## k = 1 clusters; Cases in matched pairs: 65.01 %
## k = 2 clusters; Cases in matched pairs: 96.05 %
## k = 3 clusters; Cases in matched pairs: 97.07 %
## k = 4 clusters; Cases in matched pairs: 96.78 %
## k = 5 clusters; Cases in matched pairs: 96.34 %
## k = 6 clusters; Cases in matched pairs: 97.07 %
## k = 7 clusters; Cases in matched pairs: 96.19 %
## k = 8 clusters; Cases in matched pairs: 96.49 %
## k = 9 clusters; Cases in matched pairs: 97.07 %
## k = 10 clusters; Cases in matched pairs: 97.22 %
## k = 11 clusters; Cases in matched pairs: 96.93 %
## k = 12 clusters; Cases in matched pairs: 96.93 %
```

The first number indicated the number of clusters and the second shows the ratio of correctly matched pairs. We can see that merely two clusters give us very high ratio. For 3 clusters the number is slightly bigger and exactly the same as for 6 and even 9 clusters. The best result can be observed for 8 clusters but it is significantly more than 3 which follows the huge growth of number of partitions.

Now we will check the total within-cluster sum of squares, which is the sum of the vector of within-cluster sum of squares:

$$\sum_{k=1}^{k} V(C_k) = \sum_{k=1}^{k} \sum_{x_i \in C_k} (x_i - \mu_k)^2.$$

Our intuition tells us that the more clusters we have, the smaller is this number. We will also show the between-cluster sum of squares on the same plot.

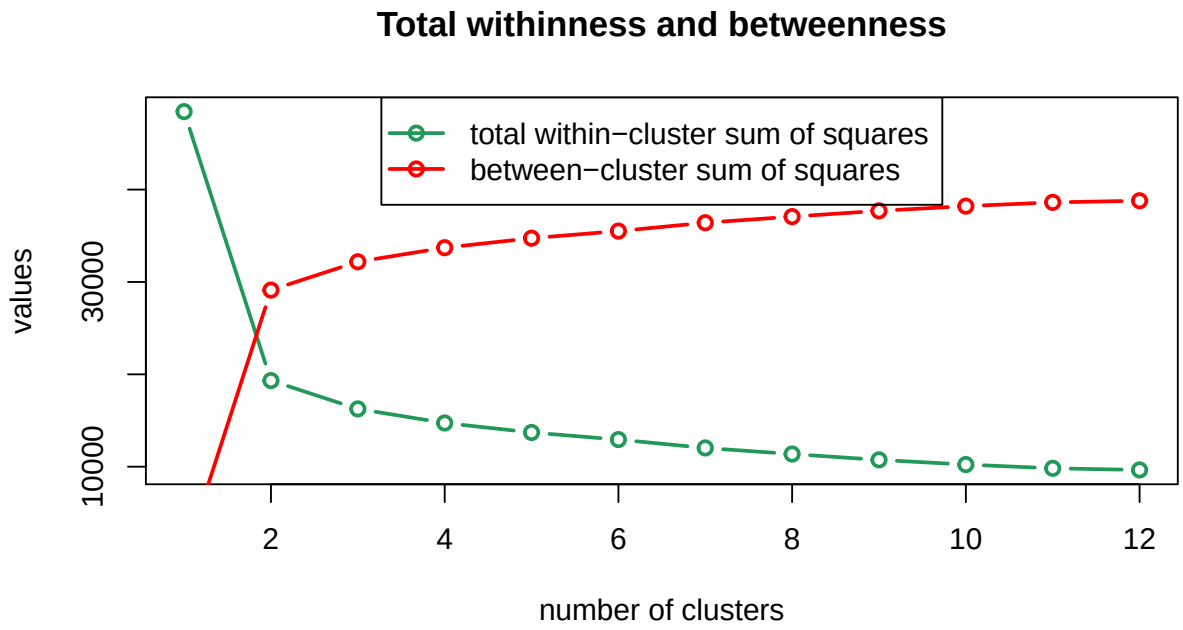


Figure 1: Sum of squares within clusters and between clusters with respect to number of clusters.

As we suspected, the bigger number of clusters is associated with the smaller value of total within-cluster variation. Starting from the very beginning the differences between the next values are getting smaller and smaller. So in our analysis we will consider two cases – the division into 2 and 3 clusters.

Creating a 2D plot showing the results of clustering for our data set might be complicated, due to the

fact that two features might take the same values for multiple observations and there are only 10 values for each feature. We have come up with an idea to show the results on 3D scatter plot. We can get $10 \cdot 10 \cdot 10$ different combinations within the values of 3 variables. Of course the part of them do not exist in our data set and some points might be duplicated, but the visualization is more clear than for 2D scatter plot. The chosen variables are these 3 that best classified our model in the Classification part of the previous project [2]. Please note that 3D scatter plot is here only for better performance of clustering and understanding the results.

The alternative to show obtained clusters is to use the observation index. Of course such approach might perturb the shapes of clusters, but as we are aware that higher values indicate malignant cancer, it is easier to spot the differences between clusters that way and no observations will overlap on the plot.

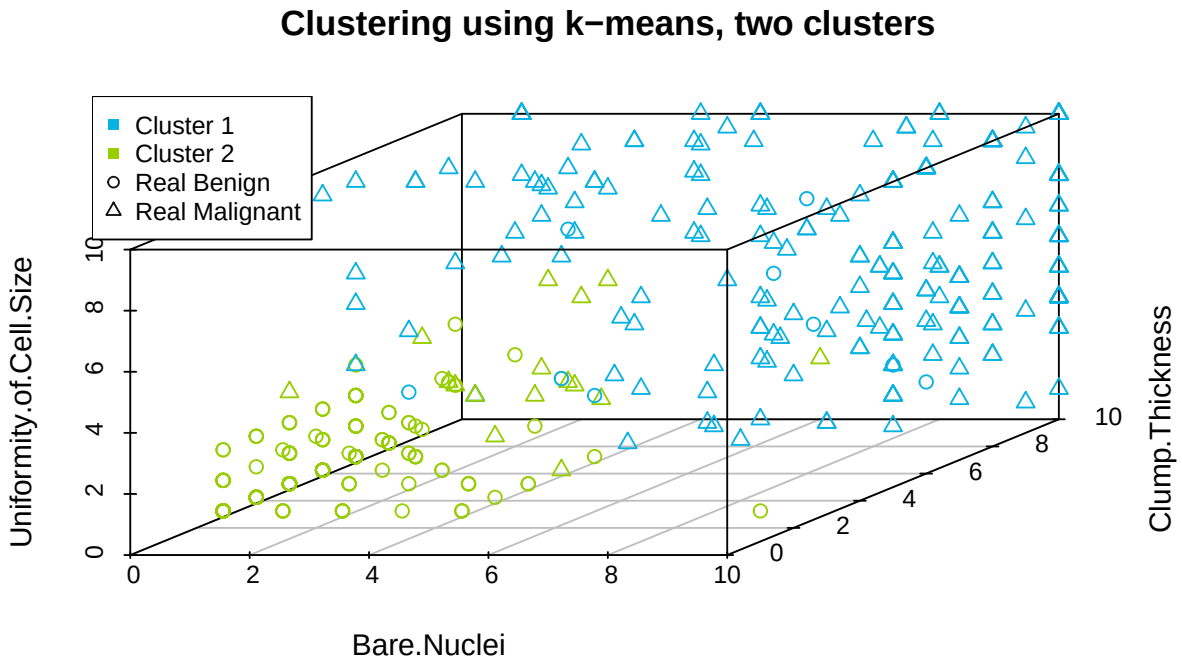


Figure 2: Cluster membership visualization.

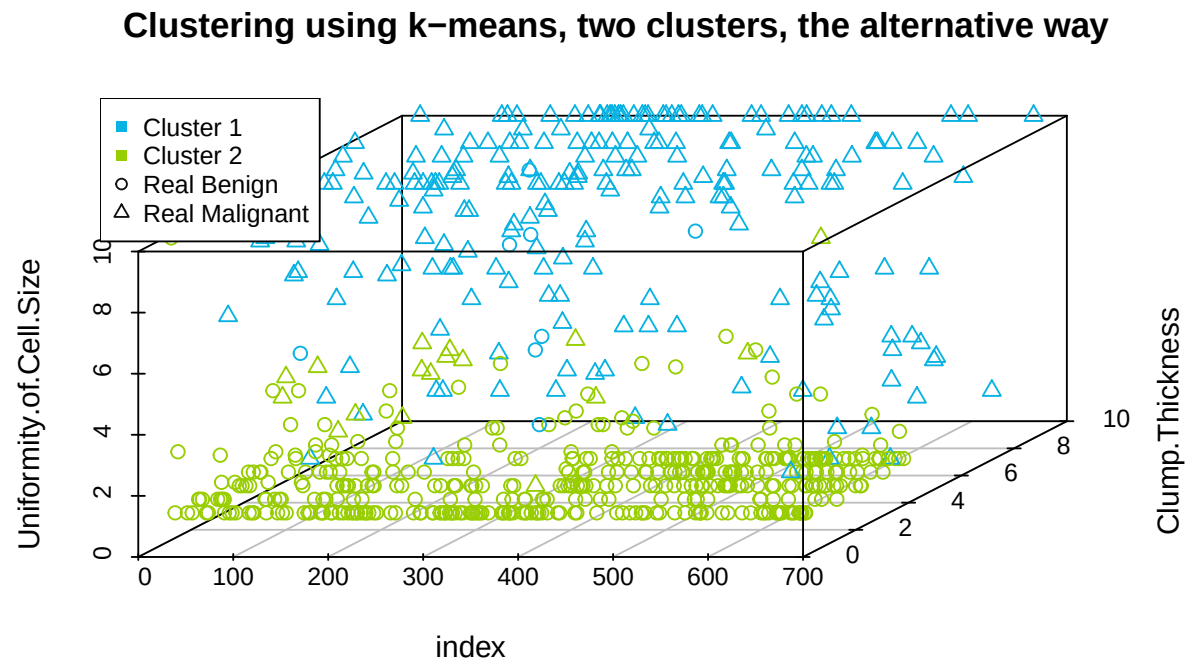


Figure 3: Cluster membership visualization.

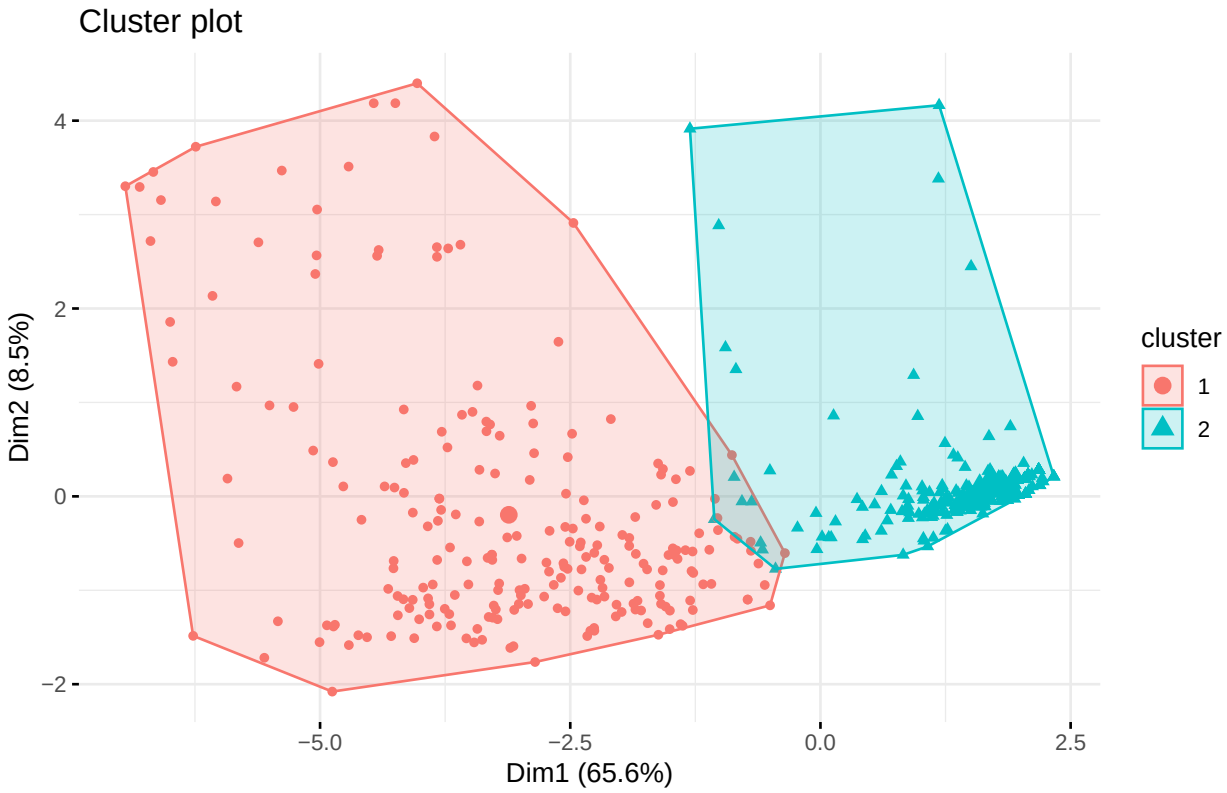


Figure 4: Visualization of cluster membership in principal components space.

| | Benign | Malignant |
|---|---|---|
| 1 | 9 | 222 |
| 2 | 435 | 17 |

The legend explains the denotation. In the figure 2 and 3 the coloured squares indicate the colours of the clusters, square shape was chosen on purpose not to be misleading with the symbols used in a plot. The empty shapes in the legend denote the real characteristics of cancer. Although the visualization in figure 3 shows all the observations (they do not overlay as in figure 2), it perturbs the cluster shape totally, so we decide not to continue showing obtained clusters in that way. Figure 4 was generated using `fviz_cluster()` function from `factoextra` package. The axes represent the dimensions obtained automatically using principal components.

As we can see in the pictures and in the table (columns indicate true classes and rows correspond to clusters), most of the true benign cancers belong to one cluster and most of the malignant belong to the other one. Now let us have a look at the partition into 3 clusters.

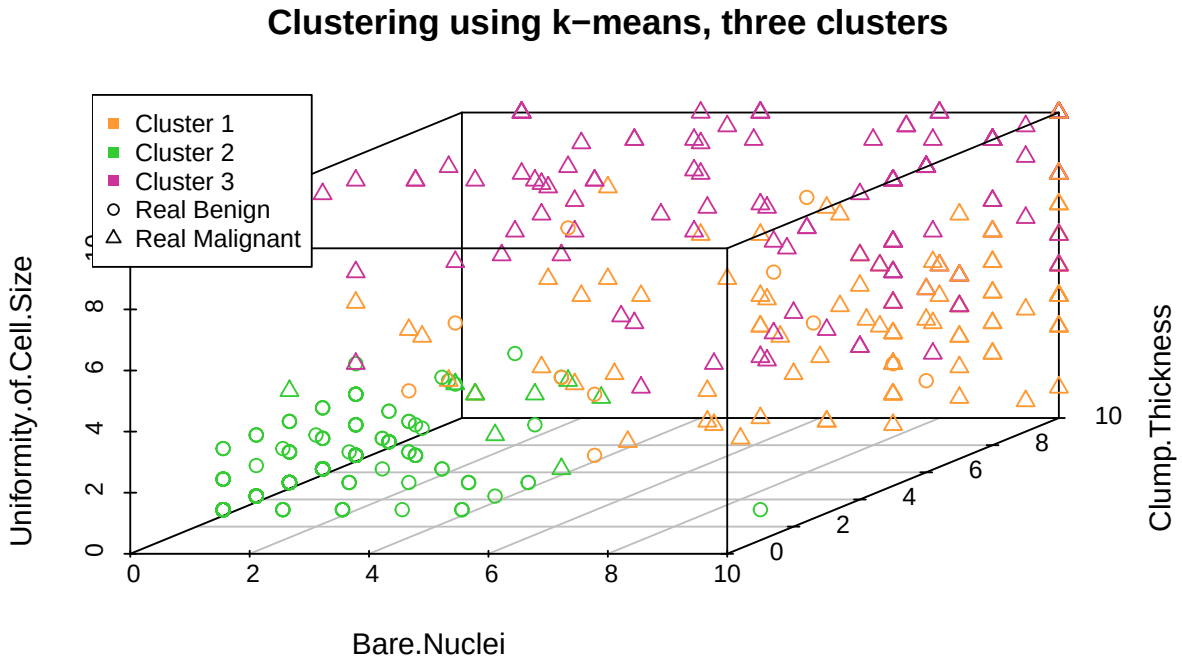


Figure 5: Cluster membership visualization.

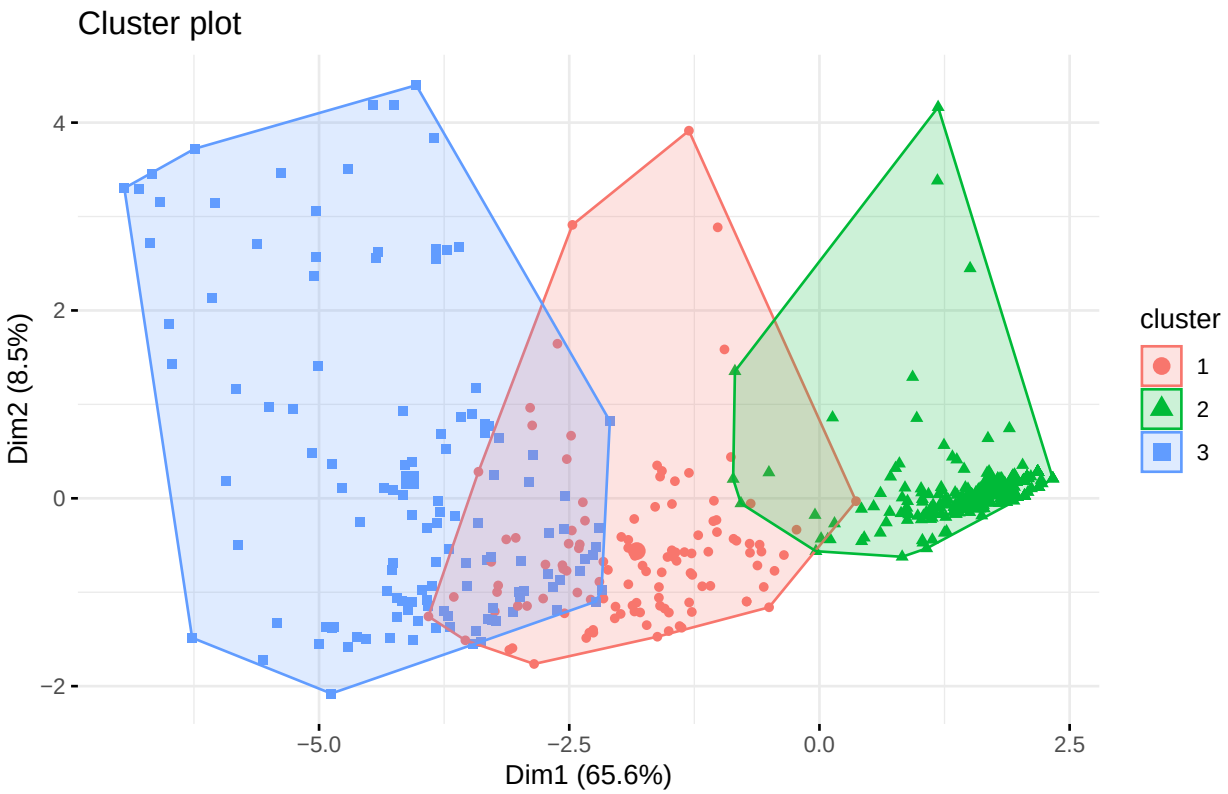


Figure 6: Visualization of cluster membership in principal components space.

| | Benign | Malignant |
|---|---|---|
| 1 | 11 | 104 |
| 2 | 433 | 9 |
| 3 | 0 | 126 |

As we can see, there is one cluster which contains observations of only one true type – malignant. In the remaining two clusters there are still a few observations assigned incorrectly.

### 3.1.2 Mini-batch-$k$-means method

Mini-batch-$k$-means method is a modification of $k$-means method. To optimize the objective function, it uses small batches from random data samples unlike in $k$-means method where the whole data set is used. The method returns the output in average more than twice as fast as the classical $k$-means. First we have to convert our data frame to matrix. Let's compare the results for both cases – 2 and 3 clusters. The algorithm will be run 5 times with different centroid seeds. The batch size was checked by us and the number 7 was chosen based on the results.

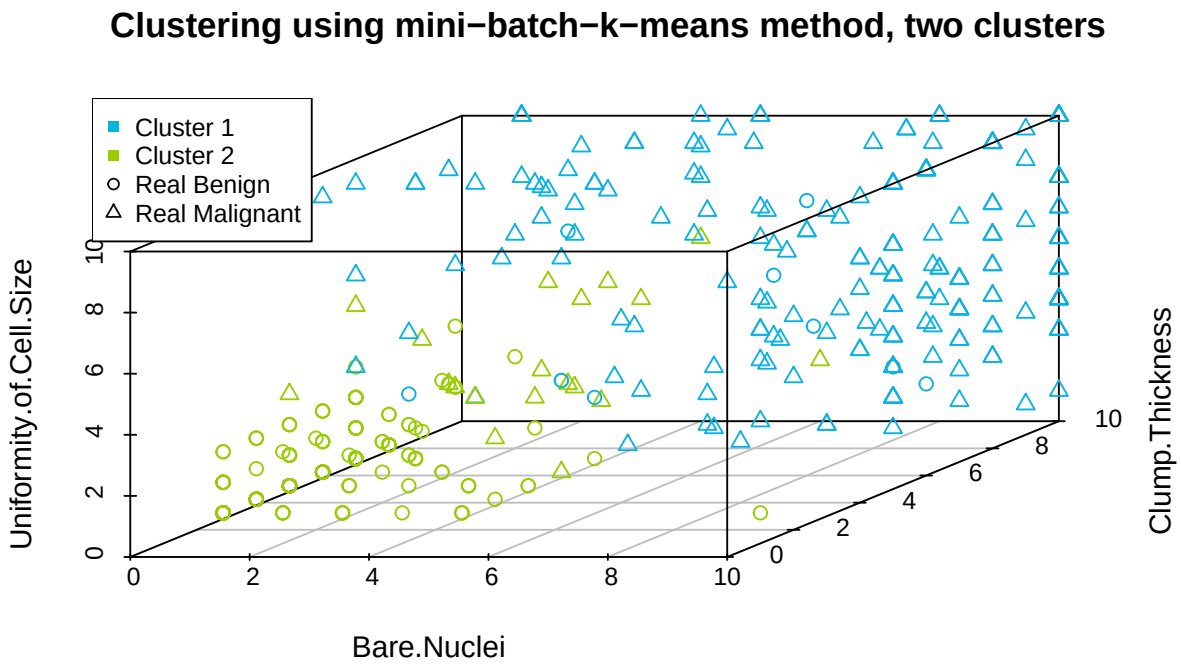


Figure 7: Cluster membership visualization.

| | Benign | Malignant |
|---|---|---|
| 1 | 9 | 219 |
| 2 | 435 | 20 |

We can see that the ratio of cases in matched pairs is a little bit lower than for classical $k$-means method. Let us check what happens for 3 clusters. Here the best ratio of matched pairs can be observed for batch size equal 17.

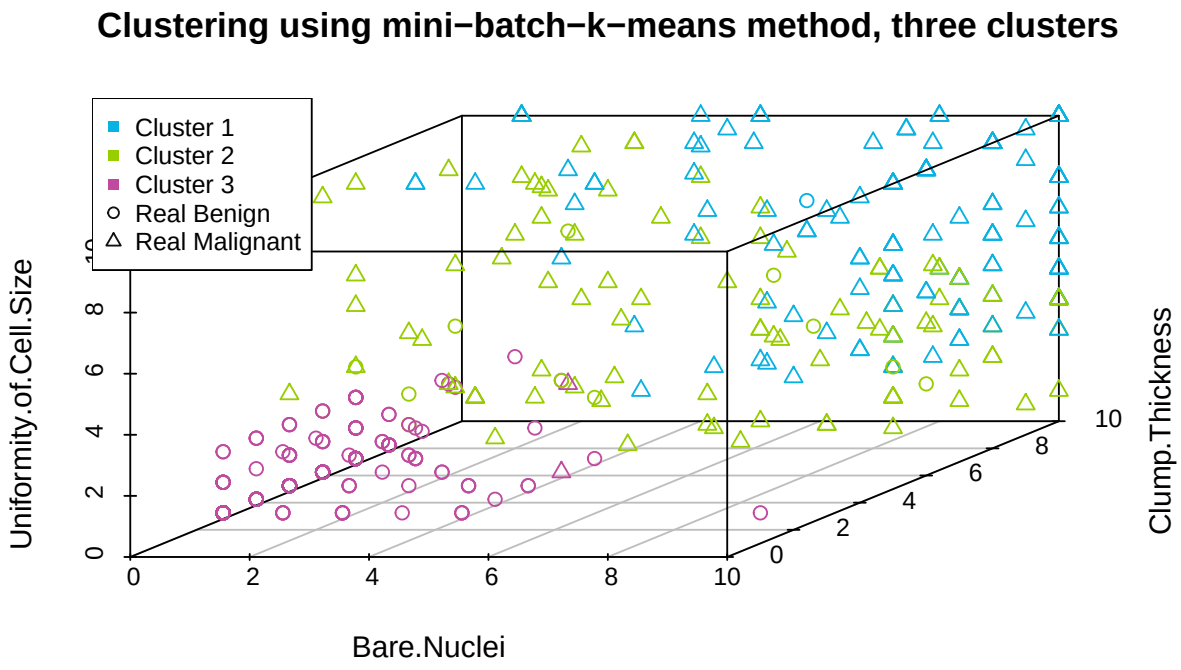


Figure 8: Cluster membership visualization.

| | Benign | Malignant |
|---|---|---|
| 1 | 1 | 142 |
| 2 | 13 | 94 |
| 3 | 430 | 3 |

As the method is very fast and efficient, we will also consider the division into 4 clusters. Here the batch size is set to 16 based on the results.

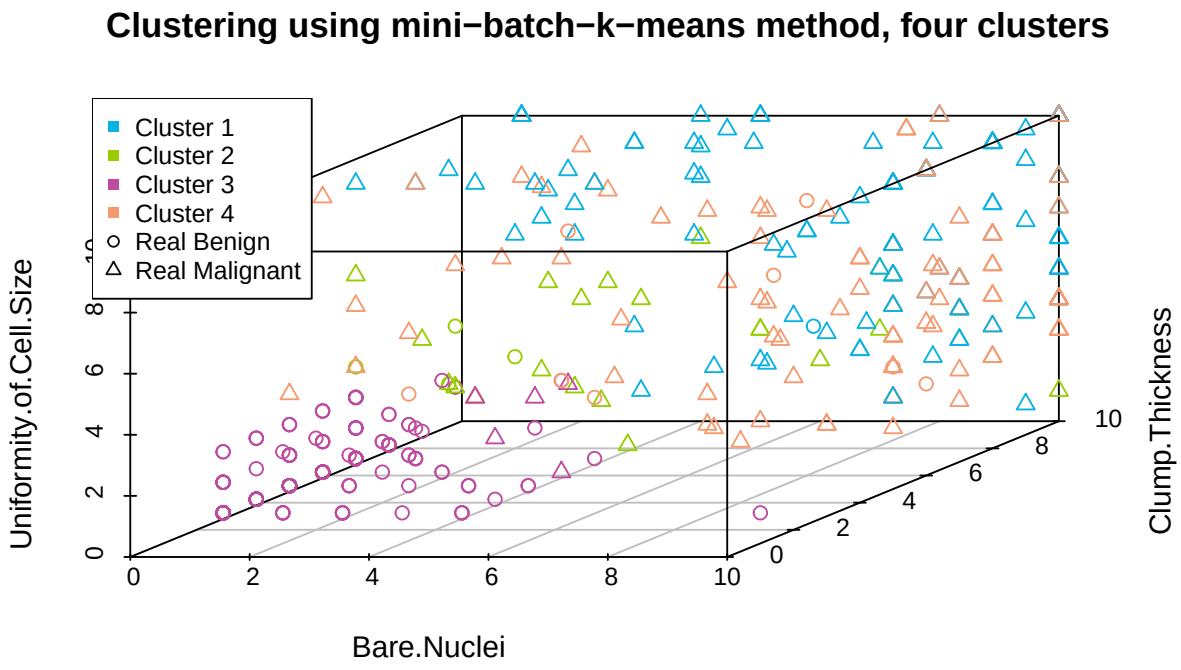


Figure 9: Cluster membership visualization.

| | Benign | Malignant |
|---|---|---|
| 1 | 1 | 115 |
| 2 | 5 | 20 |
| 3 | 430 | 5 |
| 4 | 8 | 99 |

A good observer might see that the second cluster is very small and contains only 25 values whereof 5 indicate benign cancer and the rest 20 stand for malignant cancer. It might suggest that for those observations the values of all the variables differ in such way that is it not clear what type of cancer we examine.

After experimenting with different numbers of clusters we may notice that one cluster (corresponding to benign type of cancer) is quite coherent and not affected much by introducing more clusters. No distinct groups are visible in our data and larger number of clusters does not reveal any special patterns. Thus, we will stick to the division into two clusters, because it is the most reasonable.

### 3.1.3 $k$-medians method

We are going to use a fast $k$-medians algorithm based on recursive averaged stochastic gradient algorithms (function `kGmedian()` from package `Gmedian` in R). Here the sum of norms is calculated, not the sum of squared norms like we have in $k$-means method. We do not have many outliers, but sometimes the values of our variables differ a lot so this method ensures a more robust behaviour against them. Our data frame has to be converted to matrix like previously.

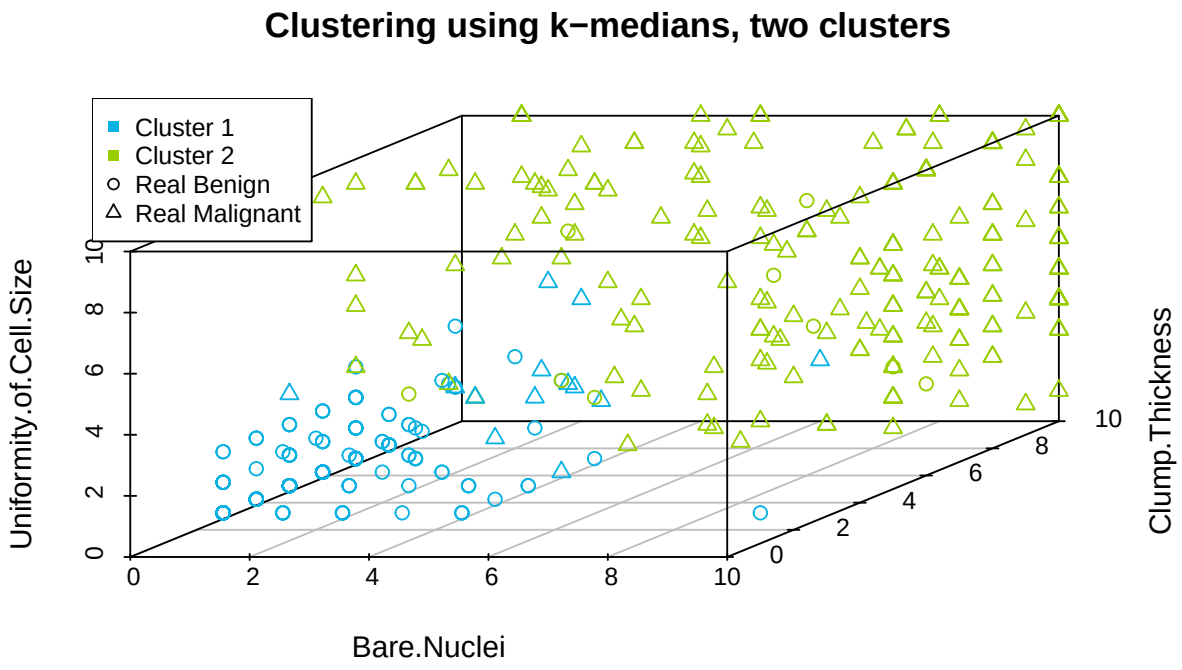


Figure 10: Cluster membership visualization.

| | Benign | Malignant |
|---|---|---|
| 1 | 435 | 14 |
| 2 | 9 | 225 |

This result seems better comparing to $k$-means. In the 'benign' cluster here we have 14 wrongly assigned observations, while standard $k$-means method mistook 18 entries.

#### 3.1.4 $k$-medoids method

$k$-medoids or partitioning around medoids (PAM) also reminds a bit $k$-means method, but PAM algorithm can be used for features of any type (also qualitative). In order to define the distance between factor observations we need to compute the dissimilarity matrix. There is a function `daisy()` in `cluster` which does it for us. We can compare here the effectiveness for both data treated as numerical and as factor.

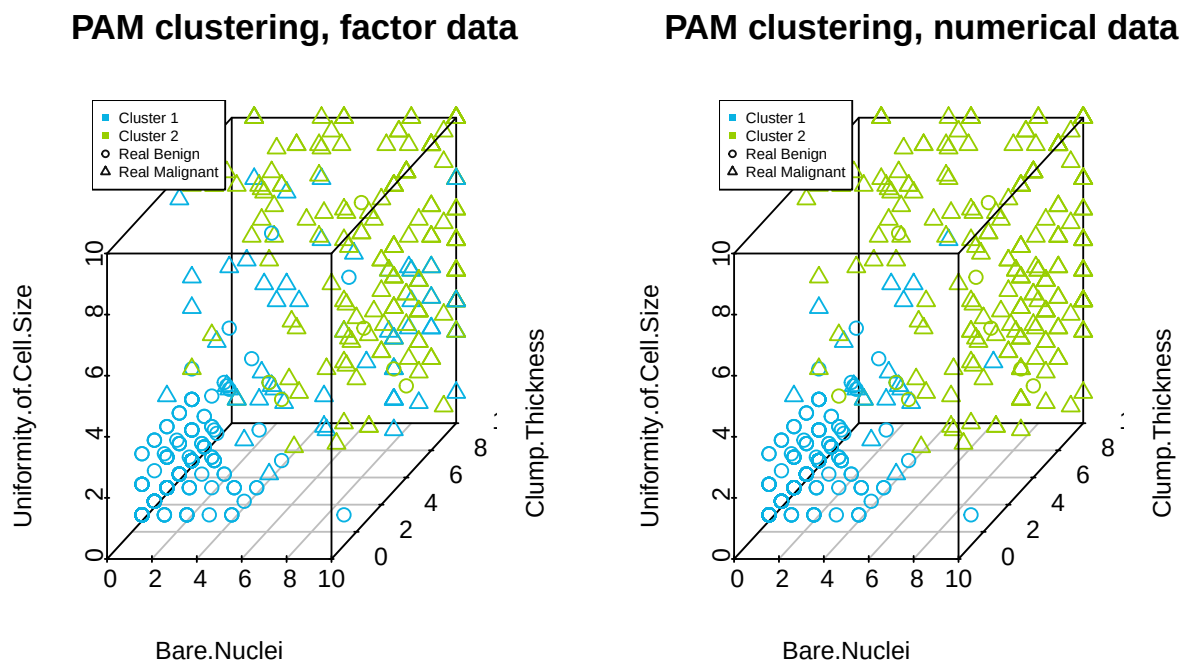


Figure 11: Cluster membership visualization.

| | Benign | Malignant |
|---|---|---|
| 1 | 438 | 61 |
| 2 | 6 | 178 |

Matched cases for PAM for factor data.

| | Benign | Malignant |
|---|---|---|
| 1 | 435 | 19 |
| 2 | 9 | 220 |

Matched cases for PAM for numerical data.

Silhouette average width (factor data): 0.3900627, Silhouette average width (numerical data): 0.597374

Basing on the plots and matching tables above, we can conclude that for this case treating data as numerical gives much better results. At the left scatter plot we can see that the computed clusters' members are not aggregated in visibly distinct groups. In addition, the silhouette index proves lower internal quality of clusters obtained for factor data.

Although we expected that PAM algorithm might work better than $k$-means, our first basic $k$-means algorithm turns out to be more effective in external indices validation (i.e. in terms of matching with true class labels). However, among all analysed partitioning methods, $k$-medians turns out to be the best. Below we present the percentage of true cases matched in pairs with computed clusters. We used function `matchClasses()`.

| algorithm | cases in matched pairs |
|---|---|
| $k$-medians | 96.63 % |
| $k$-means | 96.19 % |
| PAM | 95.90 % |
| mini-batch | 95.75 % |

We can quickly compare the internal quality of clusters obtained with different methods. The function `cluster.stats()` from `fpc` package returns a list containing many statistics useful for analyzing the intrinsic characteristics of a clustering. We considered only the cases of two clusters and $k$-means, mini-batch-$k$-means, $k$-medians and PAM methods performed on numerical data.

| | k-means | mini-batch | k-medians | PAM |
|---|---|---|---|---|
| average distance within clusters | 6.1853 | 6.2003 | 6.1749 | 6.1954 |
| average distance between clusters | 15.8619 | 15.9094 | 15.8085 | 15.8923 |
| Dunn index | 0.1580 | 0.1580 | 0.1453 | 0.1580 |
| Dunn index 2 | 1.4450 | 1.4541 | 1.4354 | 1.4515 |

We seek the minimum distance within clusters and the maximum distance between them. So in the first category the winner is $k$-medians method and in the second mini-batch-$k$-means. The first Dunn index is defined as a ratio of minimum separation to maximum cluster diameter, so the bigger the better. Here only $k$-medians method was a bit worse. The second Dunn index is a ratio of minimum average dissimilarity between two clusters to maximum average within cluster dissimilarity, so again the bigger the better and the top is mini-batch-$k$-means. However, all the methods reveal similar internal quality.

## 3.2 Hierarchical clustering

### 3.2.1 Agglomerative method

Agglomerative hierarchical clustering starts by treating each observation as a separate cluster. Then, it identifies the two clusters that are closest together, and merge the two most similar clusters. These steps continue until all the clusters are merged together.

The first important step in this analysis is to define the distance between two observations. We will again compare two approaches – treating data as numerical and calculate a simple euclidean distance using `dist()` function and creating a dissimilarity matrix for factor data.

Next step is to select the linkage method, which means from where the distance is computed. Most popular are:

- Single linkage – between the two most similar elements of the clusters,
- Complete linkage – between the two least similar elements of the clusters,
- Average linkage – between the centers of the clusters.

In the first approach we will compute the simple euclidean distance for numerical values and use `hclust()` function with specified linkage method. At the beginning, we will verify if 2 is again the optimal number of clusters. In order to do it we can compare internal quality for different numbers of clusters.

We can perform an internal quality assessment using silhouette information. For each observation $i$, the silhouette width $s(i)$ is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where
$a(i)$ = average dissimilarity between $i$ and all other points of the cluster to which $i$ belongs (if $i$ is the only observation in its cluster, $s(i) = 0$ without further calculations),
$b(i)$ = dissimilarity between $i$ and its "neighbour" cluster, i.e., the nearest one to which it does not belong.

The table below presents the average silhouette width for different number of clusters and different linkage methods.

| | Linkage method | | |
|---|---|---|---|
| no of clusters | single | complete | average |
| 2 | 0.3776 | 0.5792 | 0.5891 |
| 3 | 0.3632 | 0.4989 | 0.5503 |
| 4 | 0.3465 | 0.4941 | 0.4904 |
| 5 | 0.2362 | 0.4981 | 0.4858 |
| 6 | 0.2252 | 0.4620 | 0.4839 |
| 7 | 0.2193 | 0.4654 | 0.4836 |
| 8 | 0.2099 | 0.4680 | 0.4872 |
| 9 | 0.2007 | 0.4681 | 0.4840 |
| 10 | 0.1961 | 0.4680 | 0.4939 |

The index for the number of 2 clusters is the largest, so it is the optimum. In addition, it turned out that hierarchical clustering with single linkage method has the lowest internal quality score. Now we will perform external validation by comparing cluster memberships with true class labels. Below the matched classes tables are presented.

| | Benign | Malignant |
|---|---|---|
| 1 | 444 | 238 |
| 2 | 0 | 1 |

Matched cases for single linkage method.

| | Benign | Malignant |
|---|---|---|
| 1 | 439 | 47 |
| 2 | 5 | 192 |

Matched cases for complete linkage method.

| | Benign | Malignant |
|---|---|---|
| 1 | 436 | 31 |
| 2 | 8 | 208 |

Matched cases for average linkage method.

As we can see, hierarchical clustering based on single linkage does not provide a good division into two clusters – one cluster is of size 1. External quality assessment is pretty consistent with internal one. The best result comes from average linkage, so we can have a closer look at its visualization. (However, it is still worse than for partitioning methods.)

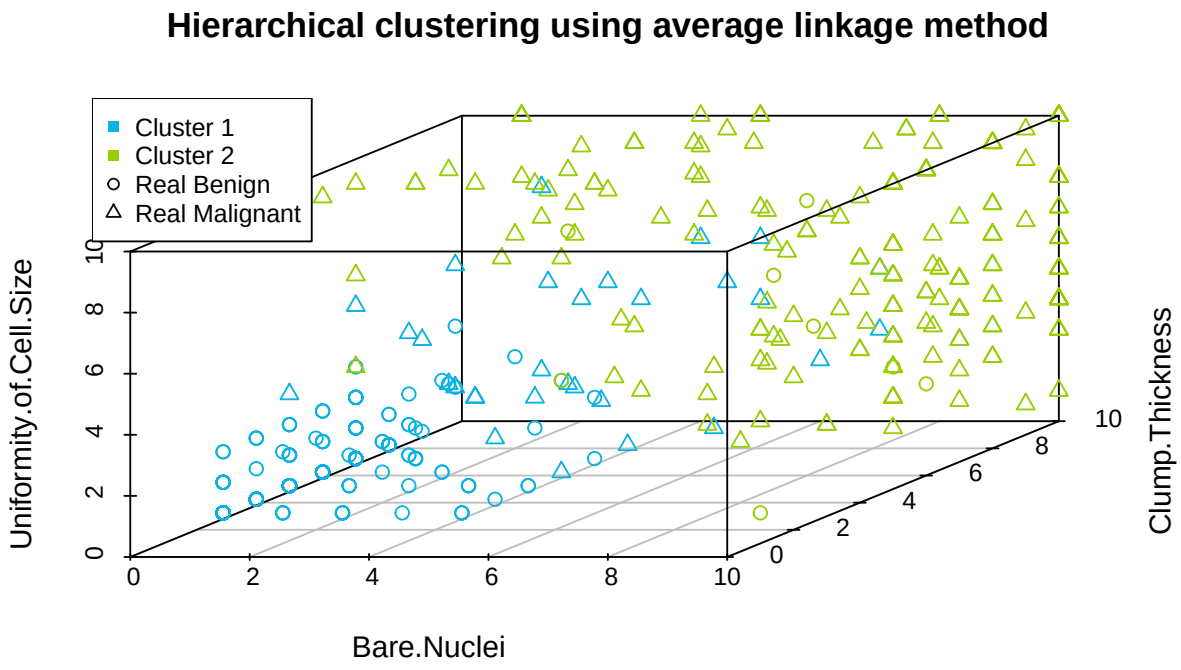


Figure 12: Cluster membership visualization.

Below there is a silhuette plot presented. High value of silhouette width indicates correct assignment. In our case there is a large number of observations assigned to the first cluster with a high confidence. Bearing in mind characteristics of our data, it can correspond to benign cancer observations, when all of the features take values close to 1. Second cluster is more problematic and not so coherent. In both clusters we can find observations which are considered as incorrectly assigned.

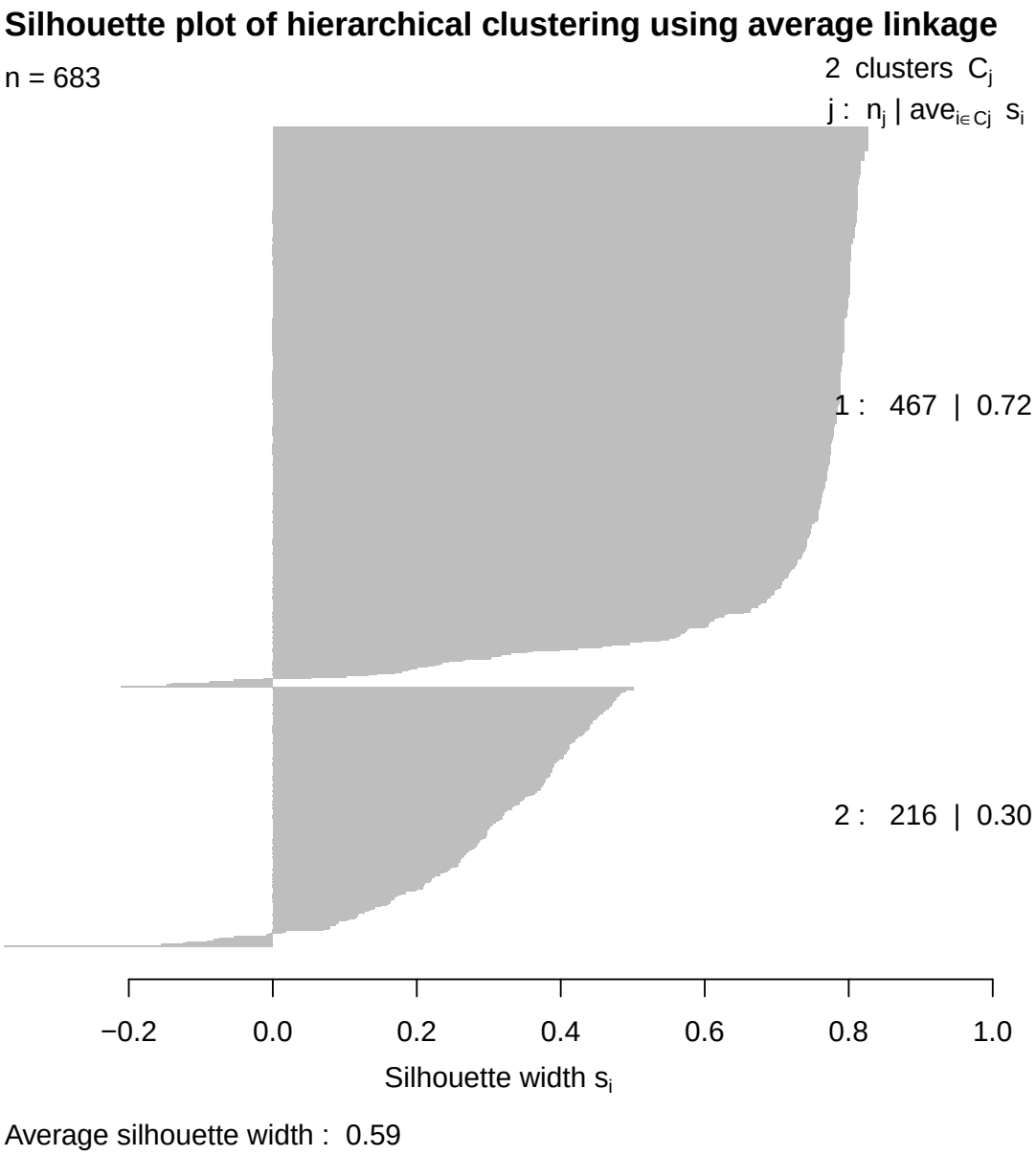


Figure 13: Silhouette plot.

Let us check whether switching to factor variables and using a dissimilarity matrix will improve the results. We will use `agnes()` function which computes agglomerative hierarchical clustering. It is very helpful to look at dendrograms. However, since we have over 600 observations, for which the dendrograms would be unreadable, we will present them for chosen 100 entries. All further computations will be done for complete data.

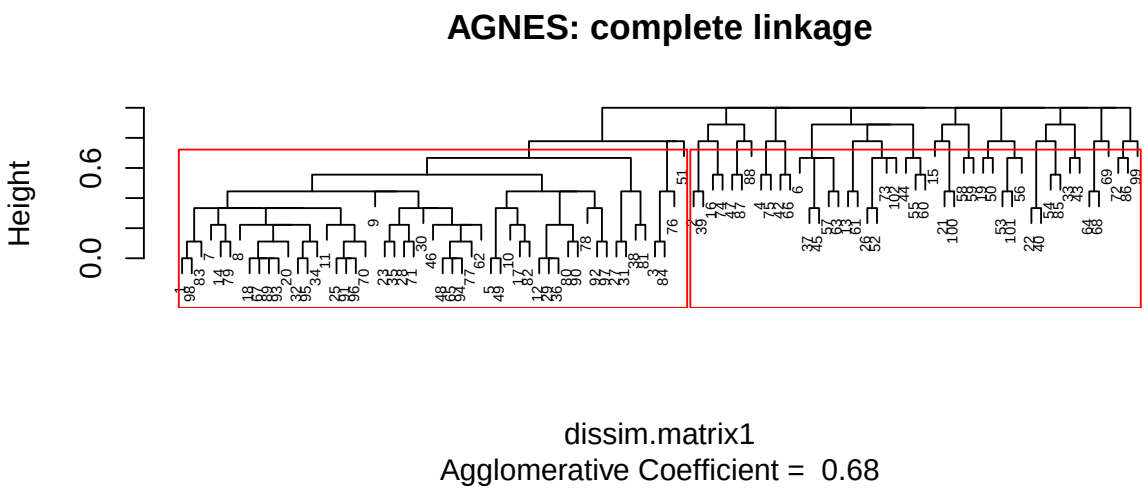


Figure 14: Dendrogram of AGNES using complete linkage method.

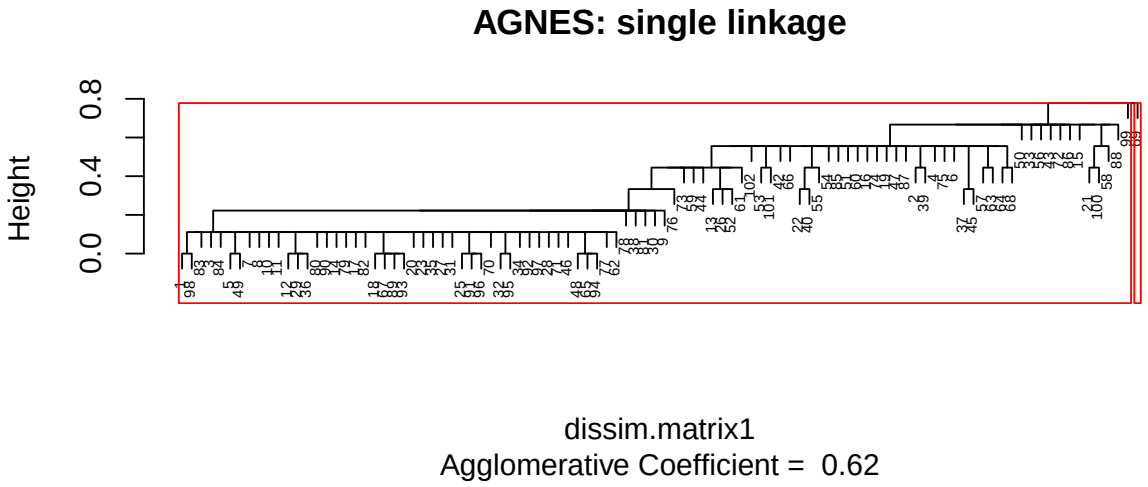


Figure 15: Dendrogram of AGNES using single linkage method.

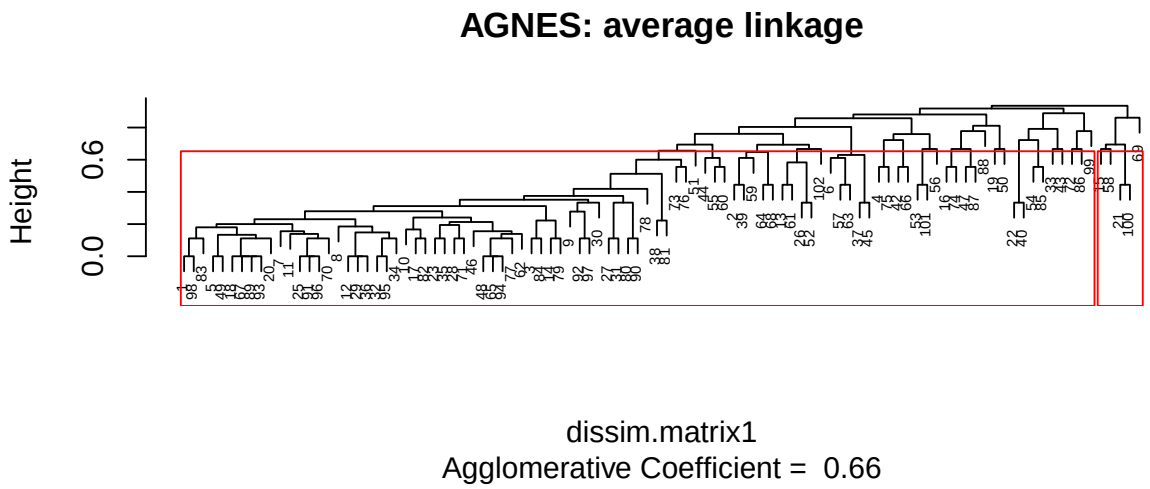


Figure 16: Dendrogram of AGNES using average linkage method.

We marked with red rectangles the regions assigned to one cluster. Only complete linkage option provides a fair cluster split. Let's check their quality according to true classes.

| | Benign | Malignant |
|---|---|---|
| 1 | 316 | 1 |
| 2 | 128 | 238 |

Cluster 1 contains only observations corresponding to benign cancer and one malignant cancer. However, in the second cluster we have both types mixed. Let's check whether if we set the number of clusters to 3, we will separate the mixed observations.

| | Benign | Malignant |
|---|---|---|
| 1 | 316 | 1 |
| 2 | 2 | 1 |
| 3 | 126 | 237 |

Unfortunately, chosen algorithm didn't work that well. It was very surprising for us that changing a distance measure and using slightly different function (but both computing agglomerative nesting) changed the results dramatically.

Keeping in mind the purpose of our analysis – detecting a separation pattern in order to categorize the cancer, we will not assess the internal quality, since the validation referring to external information (actual class membership) already shows its poorness.

### 3.2.2 Divisive method

In contrast to agglomerative nesting, divisive analysis (DIANA) works in a top-down manner. At each step of iteration, the most heterogeneous cluster is divided into two. The process is iterated until all objects are in their own cluster.

Since in the past analysis we obtained better results treating our data as numerical and calculating simple euclidean distance between observations, now we will also use DIANA algorithm on numerical data.

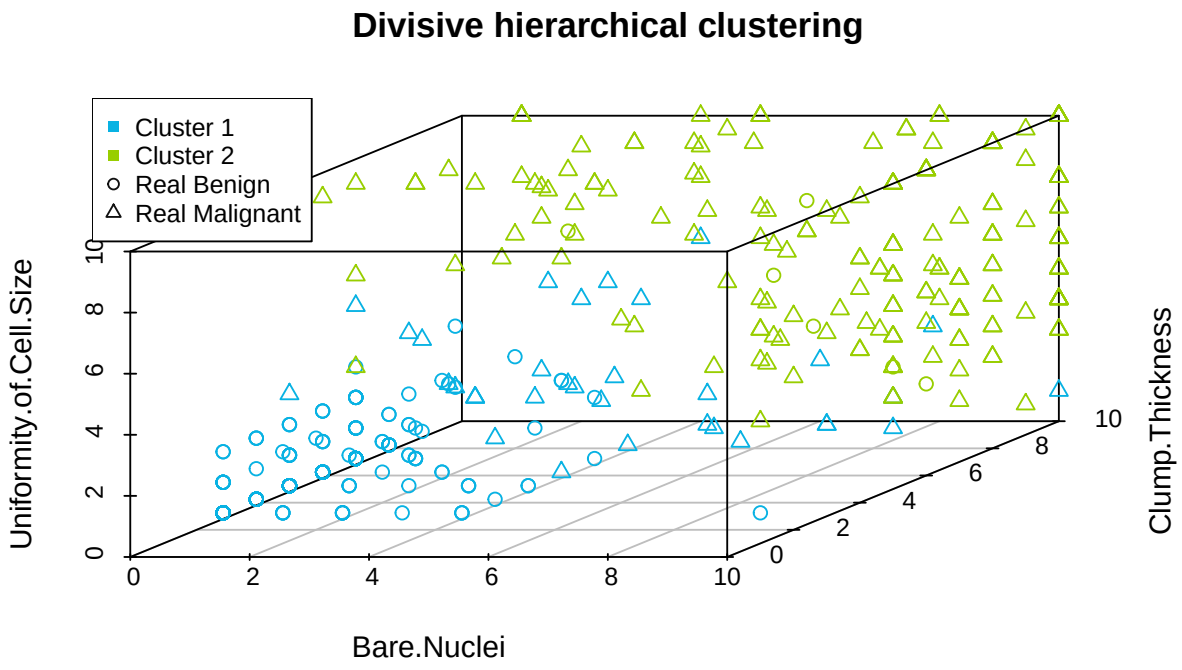


Figure 17: Cluster membership visualization.

| | Benign | Malignant |
|---|---|---|
| 1 | 438 | 34 |
| 2 | 6 | 205 |

Again, this method does not pass well the external indices validation. The partition agreement is similar to the one from agglomerative nesting with average linkage (on numerical data) and not very satisfying. Cluster 1 visibly invades not its region.

## 3.3 Other methods

### 3.3.1 Density based clustering: DBSCAN

DBSCAN is the abbreviation of Density Based Spatial Clustering of Applications with Noise. The algorithm is based on k-dimensional tree and was proposed in the year we (the authors of the project) were born. The method marks the outliers from low-density regions. It is worth mentioning that in 2014 the algorithm was awarded the test of time award at the leading data mining conference KDD [6] so it would be highly inappropriate not to try to apply the method to our data set. It estimates the density around each observation based on the chosen neighbourhood. After that the borders of the clusters are specified.

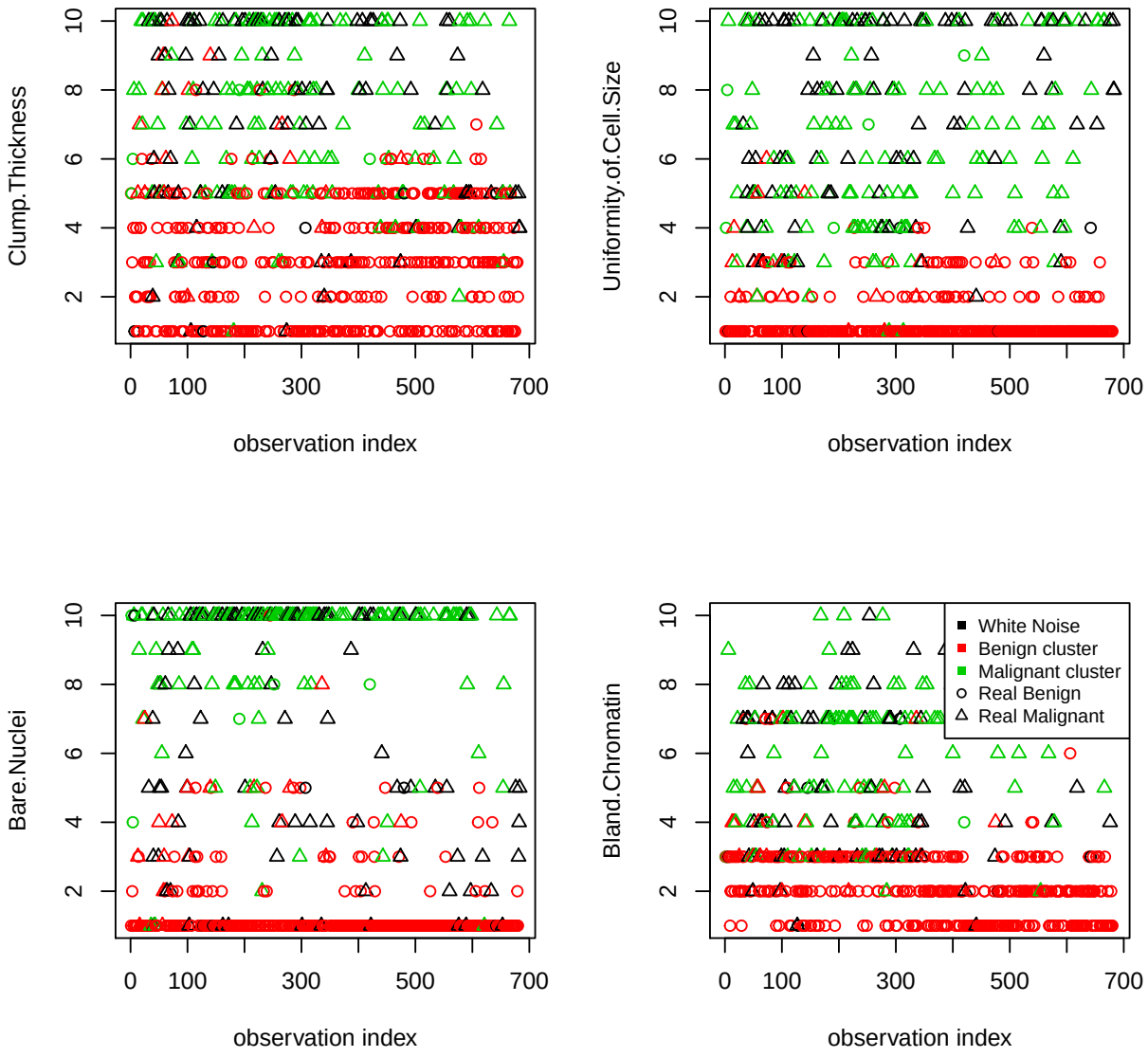


Figure 18: DBSCAN clustering results

| | Benign | Malignant |
|---|---|---|
| 0 | 6 | 94 |
| 1 | 433 | 18 |
| 2 | 5 | 127 |

This way of visualization allows us to show the result of DBSCAN method. We chose 4 variables to present the differences between the clusters on the plots. Black points are considered white noise. The remaining observations were grouped into two clusters and we can really see the separation between them in all the pictures. Observations which take high values of a characteristic are grouped together and correspond to true malignant clump, and observations which take low value of a characteristic are in general corresponding to benign clump. The index of the variable is used just to avoid overlapping and also show that the value of one separate variable can more or less exhibit the differences itself. In the table we can see that true benign cancers lie mostly in one cluster. Only six observations were considered as white noise by the algorithm

(row 0 corresponds to white noise observations). Malignant ones are again difficult to group. Although the error connected with 'malignant' and 'benign' clusters separation is not large, the number of observations considered as white noise is significant. We can clearly see them in the picture below.

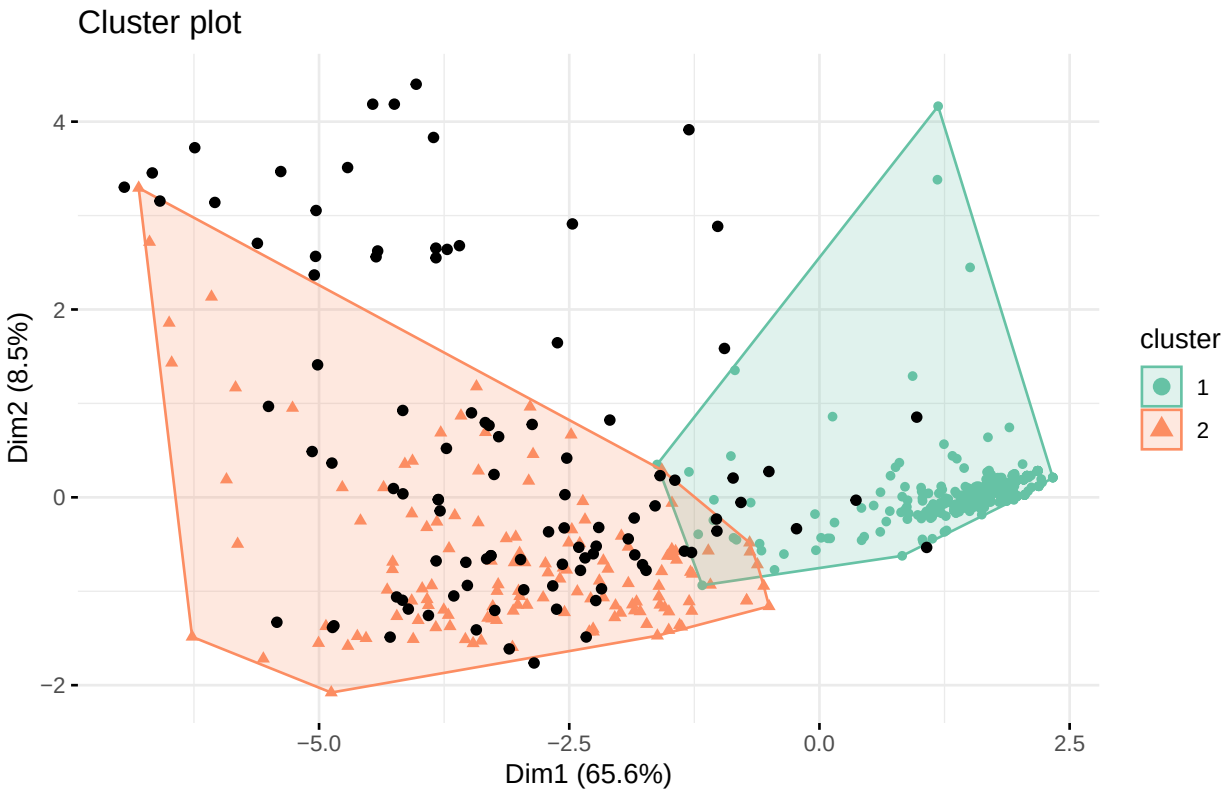


Figure 19: Visualization of cluster membership in principal components space for DBSCAN method.

# 4 Dimensionality reduction

## 4.1 PCA

In this project we will use a popular method of dimensionality reduction – principal component analysis (PCA). There are three goals when finding lower dimensional representation of features:

- find linear combination of variables to create principal components,
- maintain most variance in the data,
- principal components are uncorrelated (orthogonal).

The first important application of PCA is a visualization of multidimensional data in 2D space. We already took advantage of it presenting our clustering results in figures 4, 6 and 19. Below we can see that visualizing the data in 2D for chosen pairs of variables is unreadable, not to mention points' overlapping.

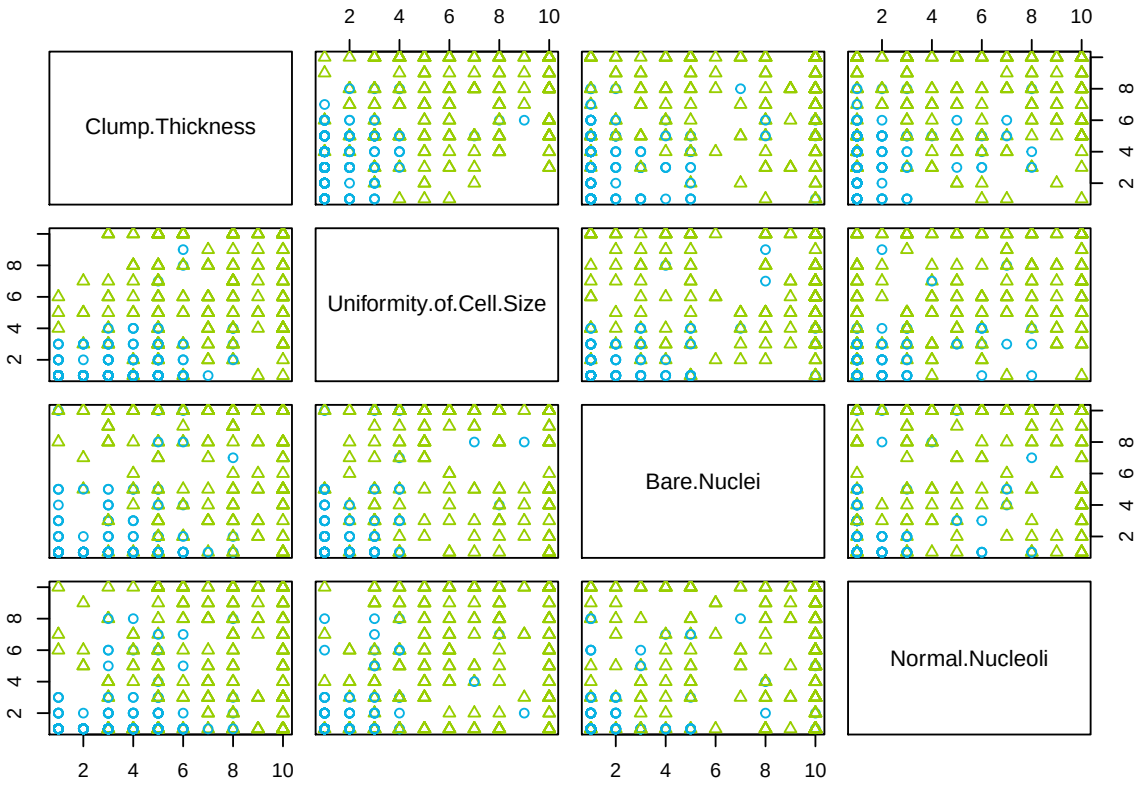


Figure 20: Data visualization for different variable pairs, coloured by class.

Of course this method works only for numerical data. In order to efficiently perform PCA, we should take care of similar properties of each variable (mean and standard deviation), since is it a variance maximizing exercise. In R function `prcomp()` there is a parameter `scale` and `center` which, if `True`, will control it for us.

Let us perform principal component analysis and see the variance explained by each component.

| | PC1 | PC2 | PC3 | PC4 | PC5 | PC6 | PC7 | PC8 | PC9 |
|---|---|---|---|---|---|---|---|---|---|
| Standard deviation | 2.4302 | 0.8751 | 0.7342 | 0.6798 | 0.6169 | 0.5501 | 0.5427 | 0.5107 | 0.2973 |
| Proportion of Variance | 0.6562 | 0.0851 | 0.0599 | 0.0513 | 0.0423 | 0.0336 | 0.0327 | 0.0290 | 0.0098 |
| Cumulative Proportion | 0.6562 | 0.7413 | 0.8012 | 0.8526 | 0.8948 | 0.9285 | 0.9612 | 0.9902 | 1.0000 |

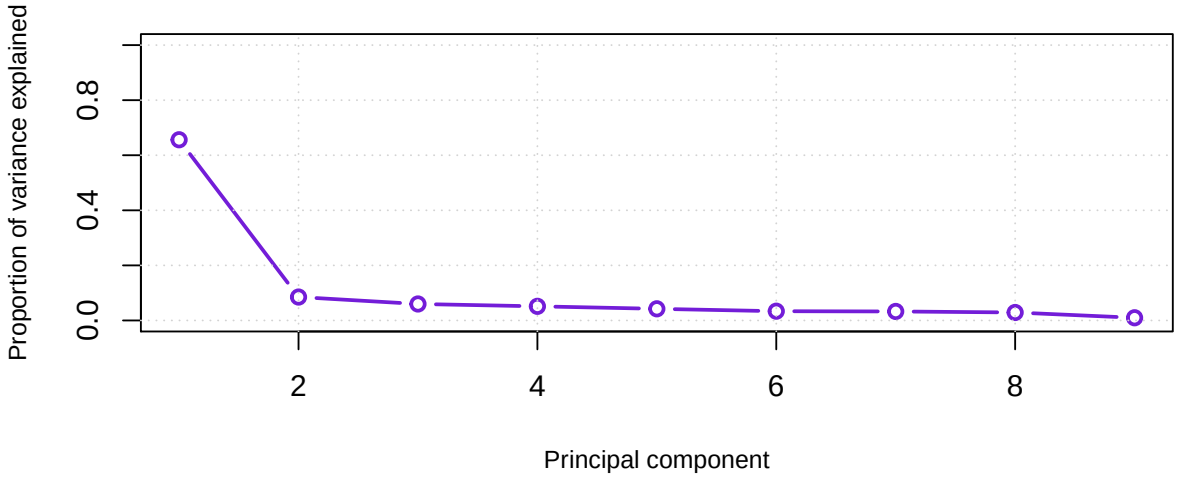


Figure 21: Proportion of variance explained for consecutive principal components.

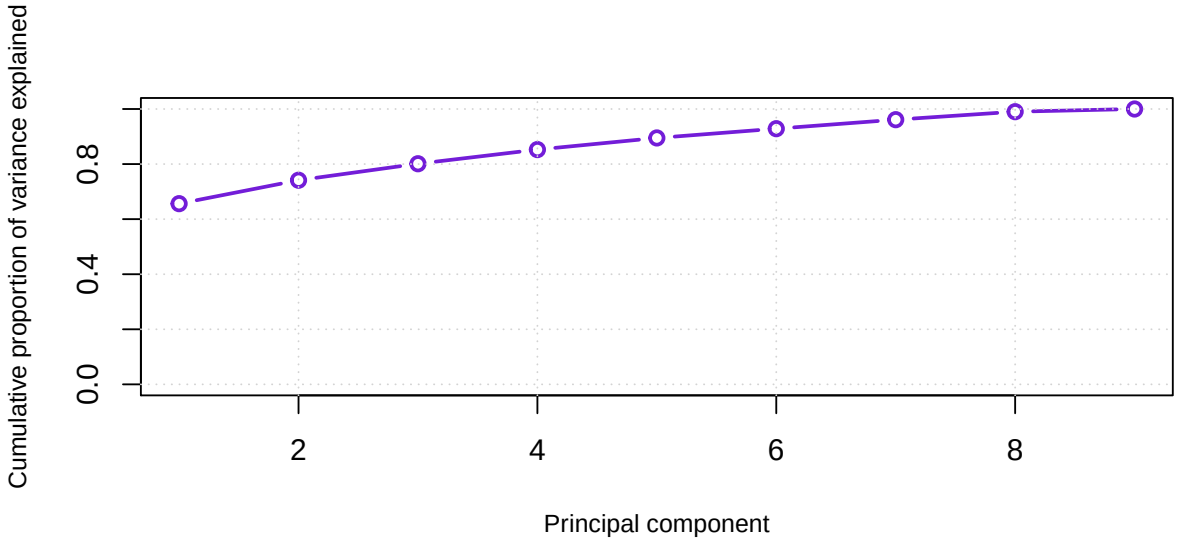


Figure 22: Cumulative proportion of variance explained for consecutive principal components.

We can see that 80% of the variance is explained by three first principal components. First two components explain 74% of the variability. Let us look at how our true classes are visible at the new space.

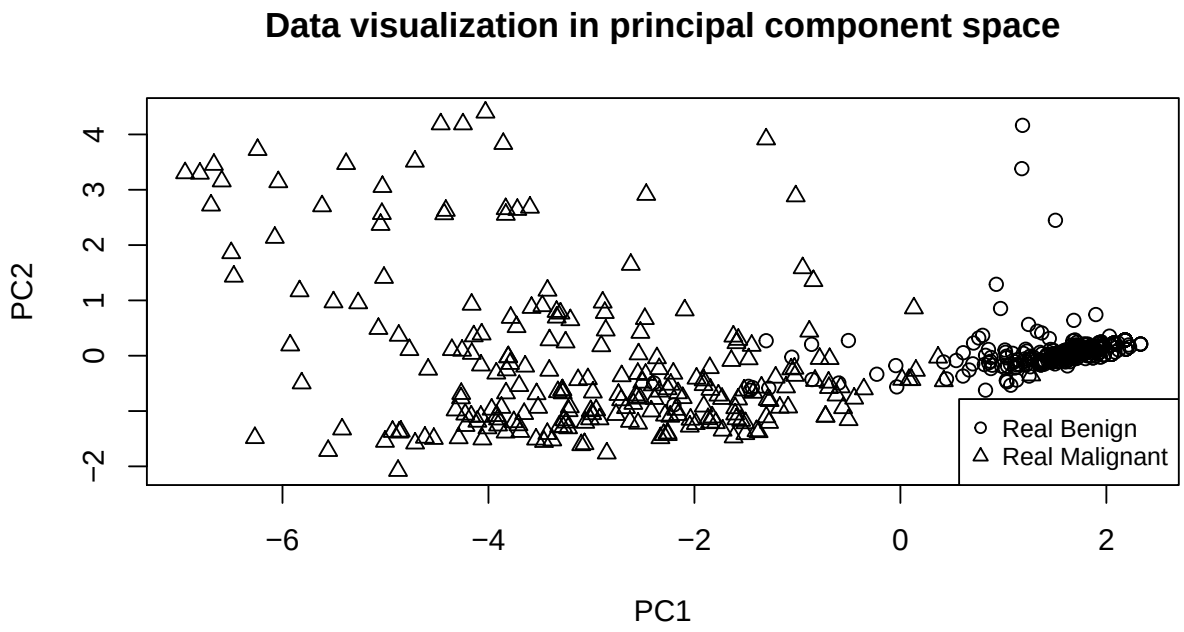


Figure 23: Data visualization in 2D in principal component space.

As we can see, it is sufficient to present the data on 2D plot and have a clear distinction between two classes. New features are not intuitive at all and have no direct interpretation in reality, so it is hard to describe such an illustration. However, this quick insight has let us assume that PCA is a great tool to preprocess data before performing classification or clustering.

## 4.2 Classification in PC space

We can take advantage of PCA to perform classification on our data. For this moment we will use a data set containing true class labels in order to train the model. The remaining 9 features are treated with `prcomp()` function. The data frame is split into the training and testing set.

First model we want to analyze is $k$**-nearest neighbours**. Of course, since we have 9 independent variables, we obtain 9 principle components, but we keep in mind that 3 of them is enough to explain 80% of the variance. We will use the algorithm for both full data frame and only the main components performing 5-fold cross validation and inspecting accuracy measures.

We obtained $k = 5$ nearest neighbours to be the optimal number. We will use function `ipredknn()` firstly for all components.

| | Mean | Variance |
|---|---|---|
| Misclassification error | 0.0351 | 0.00004 |
| Sensitivity | 0.9531 | 0.00020 |
| Specificity | 0.9707 | 0.00010 |
| Precision | 0.9458 | 0.00032 |

Accuracy measures for full model.

| | Mean | Variance |
|---|---|---|
| Misclassification error | 0.0410 | 0.00012 |
| Sensitivity | 0.9404 | 0.00128 |
| Specificity | 0.9686 | 0.00014 |
| Precision | 0.9411 | 0.00054 |

Accuracy measures for simplified model (3 principal components).

We can now take a look into the first part of our project to compare accuracy measures (section 5.1). Using only 3 PC results in a little bit worse accuracy, but in both cases prediction based on original variables is more precise.

We will consider one more model to compare the outcome. We picked **logistic regression** (section 5.2 in the first part of project). Since the prediction is given by a number between 0 and 1, we will assign "Benign" flag to those below 0.5 and "Malignant" flag to the rest. Here are the accuracy measures:

| | Mean | Variance |
|---|---|---|
| Misclassification error | 0.0337 | 0.00029 |
| Sensitivity | 0.9442 | 0.00192 |
| Specificity | 0.9778 | 0.00040 |
| Precision | 0.9568 | 0.00165 |

Accuracy measures for full model.

| | Mean | Variance |
|---|---|---|
| Misclassification error | 0.0308 | 0.00020 |
| Sensitivity | 0.9572 | 0.00175 |
| Specificity | 0.9753 | 0.00021 |
| Precision | 0.9535 | 0.00077 |

Accuracy measures for simplified model (3 principal components).

We can notice that logistic regression is more accurate than $k$-nn when we examine only 3 principal components. However, unfortunately again PCA did not boost our classification model in comparison to the first part of project results.

## 4.3 Clustering in PC space

In this section we will perform unsupervised learning using chosen methods for the new features obtained with principal component analysis. Using elbow method we can again obtain 2 clusters as an optimum.


Figure 24: Total sum of squares within clusters (obtained using k-means algorithm).

Since we met the best results for $k$**-medians** clustering, we will repeat this approach now. We considered only 2 main principal components.

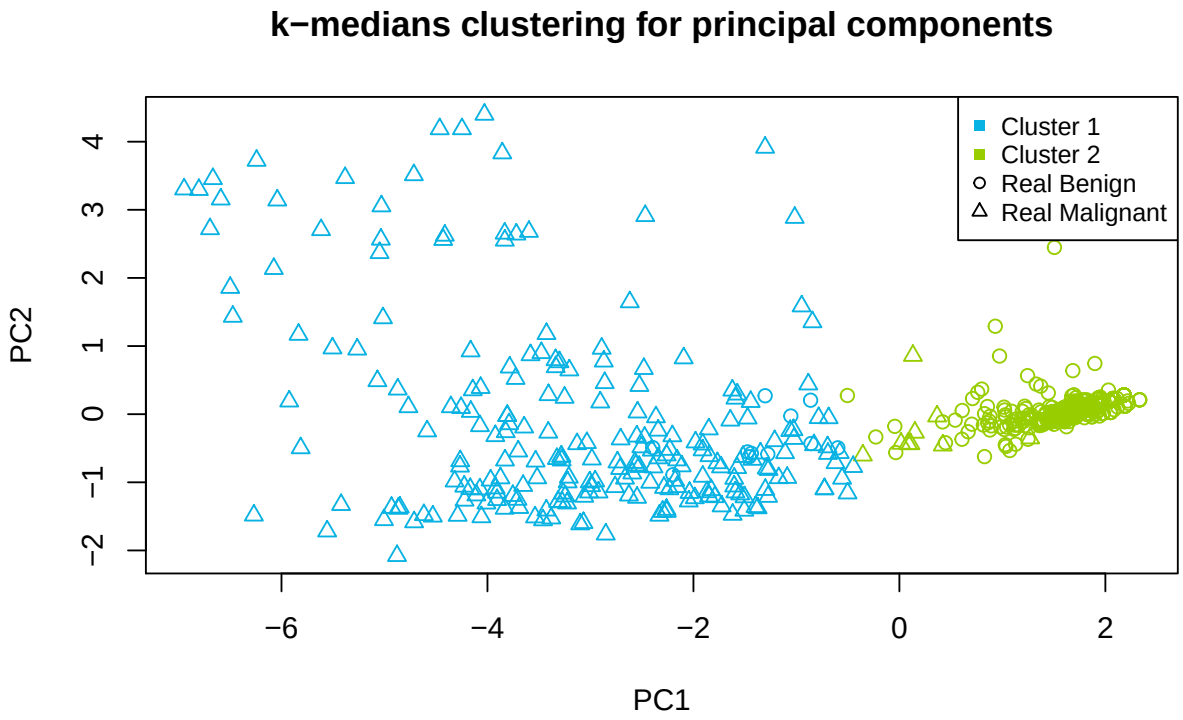


Figure 25: Cluster membership visualization.

| | Benign | Malignant |
|---|---|---|
| 1 | 11 | 230 |
| 2 | 433 | 9 |

Comparing to the previous results of $k$-medians clustering, here we have 97.07% cases in matched pairs, which is a little bit better (and we used only two features instead of nine!). Below we checked also the performance of **PAM** algorithm.

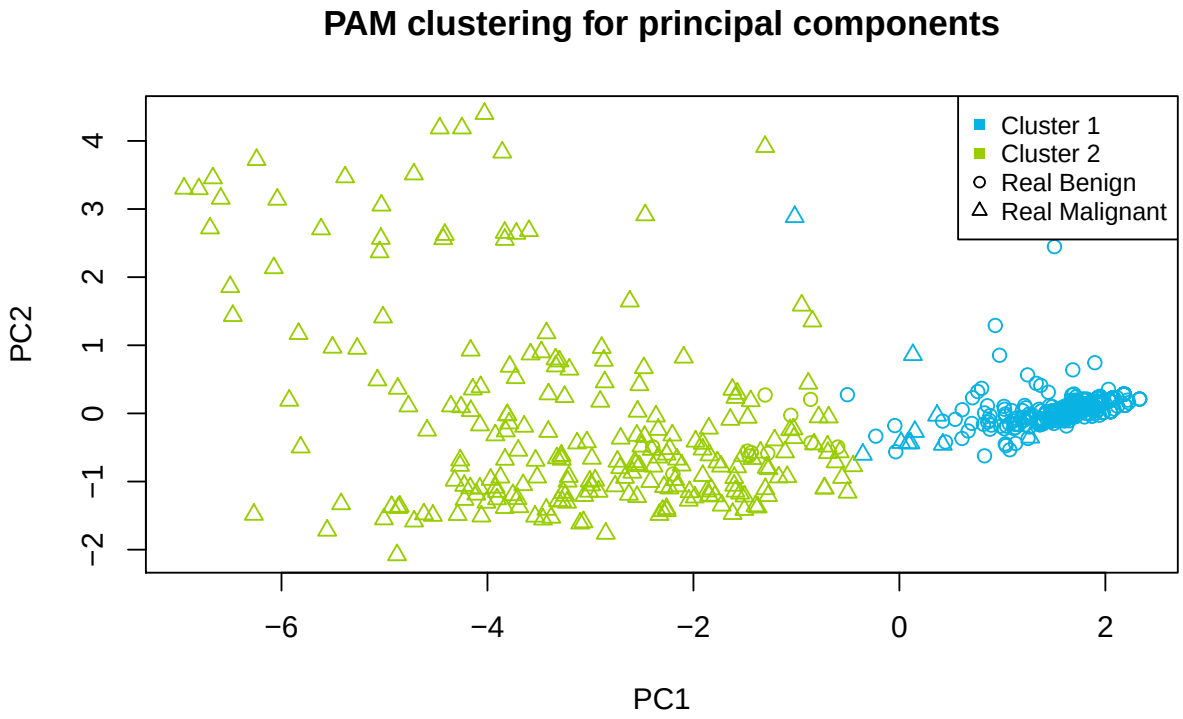


Figure 26: Cluster membership visualization.

| | Benign | Malignant |
|---|---|---|
| 1 | 433 | 10 |
| 2 | 11 | 229 |

PAM algorithm assigned only one observation differently (unfortunately incorrectly regarding the true classes). As we remember, in original variable space PAM mismatched true labels with clusters in 28 cases. Now we have only 21 mismatched cases.

# 5 Conclusions and remarks

We have come a long way to find ourselves in this place and gain significant knowledge in the field of data mining techniques, in particular classification, clustering and proper data analysis. We have penetrated multiple feature selection methods; got through well-known and less popular classification approaches; confronted various clustering algorithms thereby facing the vastness of data mining field.

In this part of the project we have focused mostly on clustering and dimensionality reduction. We started off with partitioning methods like $k$-means, mini-batch-$k$-means, $k$-medians and $k$-medoids. Then the hierarchical clustering was our target: divisive method and agglomerative method with silhouette plot and dendrogram. We have gained the knowledge regarding density based clustering (DBSCAN).

The results perpetuated us in the belief that **two** clusters is what we need and should consider. Dealing with numerical data gave us good results and satysfying internal quality of clusters. With reference to external indices validation $k$-medians algorithm seems to be the most efficient, although we suspected PAM to give better results. Considering all the criteria together, some methods match better with true classes, some exhibit better internal quality.

Within the hierarchical methods we realized that average linkage gives the best results. After focusing on DIANA we realized that hierarchical methods do not cope with our data as well as partitioning methods.

We performed dimentionality reduction using Principal Component Analysis. Although it transforms our variables into less understandable and not intuitive components, it allows us to visualize multidimensional data in 2D space. Using classification algorithms on the transformed breast cancer data turned out not to be more accurate than on original data. However, it played a bigger role in clustering. Two main principal components were enough to create more consistent with true labels clusters.

We tried hard to detect a separation pattern in order to categorize the cancer using multiple approaches and eventually we managed to do it. Data mining techniques will no longer be a secret.

# 6 Bibliography

# References


[1] Medical diagnostics: Breast Cancer Wisconsin (Original) Data Set
`http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original)`.
UCI, Machine Learning Repository, 1992.

[2] M. Kawalko, Z. Materny, *Application of data mining techniques on Breast Cancer Wisconsin data set part I* (2019)

[3] A. Zagdanski, Lecture materials for Data Mining course (2019)

[4] S. Jaiswal, DataCamp, K-Means Clustering in R Tutorial (2018)

[5] M. Pathak, DataCamp, Hierarchical Clustering in R (2018)

[6] `https://www.kdd.org/News/view/2014-sigkdd-test-of-time-award`

[7] A. Geron, *Uczenie maszynowe z uzyciem Scikit-Learn i TensorFlow* (2017)

[8] DataFlair Team, *Clustering in R - A Survival Guide on Cluster Analysis in R for Beginners!* (2019)

[9] L. Fonseca, *Clustering Analysis in R using K-means* (2019)

[10] G. Martos, *Cluster Analysis with R*