TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Jevgeni Menšenin, 153073IAPM

DEVELOPING DATA ANALYSIS BASELINE FOR SPARSE CLINICAL DATASET WITHOUT DOMAIN KNOWLEDGE

Master’s thesis

Supervisor: Martin Rebane
MSc

Tallinn 2017

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Jevgeni Menšenin, 153073IAPM

DOMEENITEADMISETA ANDMEANALÜÜSI BAASJOONE ARENDAMINE HÕREDA KLIINILISE ANDMESTIKU JAOKS

Magistritöö

Juhendaja: Martin Rebane
MSc

Tallinn 2017

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Jevgeni Menšenin

15.05.2017

Abstract

The aim of this work was to develop a data analysis baseline which could be used as a basis for sparse clinical data analysis. Clinical data is domain specific and if the expert knowledge is not accessible during analysis, applying the baseline should be possible without domain knowledge. Main purpose of developed baseline is to simplify and accelerate analysis of clinical data.

Considering the clinical data properties (such as high amount of missing values and low amount of records), developed baseline consists of 3 steps: preprocessing (data cleaning), transformation (dimensionality reduction) and data mining (clustering analysis and association rules mining). Those steps are focused on extracting valuable information from dataset and improving the quality of this information. Those steps was implemented using Python and R programming languages with open source libraries.

To test developed data analysis baseline we were provided with real clinical dataset, which was gathered during the treatment and observation of patients with cardiac diseases. Developed baseline was applied to this data without using expert knowledge for generating results (clusters, association rules), which were provided to domain expert for interpretation and evaluation.

Finally, time complexity of the baseline with bigger datasets was empirically validated. It was applied to voluminous datasets and time, used by baseline processes was measured.

Overall, baseline was developed and its application has demonstrated that it is possible to analyze clinical data without domain knowledge, however, certain steps would benefit from an expert's advice.

This thesis is written in English and is 70 pages long, including 8 chapters, 21 figures and 15 tables.

Annotatsioon

Antud töö põhiliseks eesmärgiks oli andmete analüüsi *baseline*'i arendamine, mida saaks kasutada hõredate kliiniliste andmete analüüsi baasina. Kui eksperdi arvamus ei ole saadaval analüüsi käigus, siis sellel juhul oleks võimalik *baseline*'i rakendada ilma domeeniteadmisteta. Arendatud *baseline*'i peamine eesmärk on lihtsustada ja kiirendada kliiniliste andmete analüüsi ja töötlemist.

Arvestades kliiniliste andmete omadusi (nagu suur hulk puuduvaid ja osaliselt sisestatud andmeid), koosneb arendatud *baseline* kolmest sammust: eeltöötlemine (andmete puhastamine), transformatsioon (dimensionaalsuse vähendamine) ning andmete kaevandamine (klasterdamine ning assotsiatsioonireeglite kaevandamine). Need sammud on keskendunud väärtuslike andmete leidmisele andmehulgast ja info kvaliteedi parandamisele. Realiseerimiseks kasutati Python ja R programmeerimiskeeli ja avaliku lähtekoodiga raamistikke.

Arendatud andmete analüüsi *baseline* testimiseks kasutasime tegelikku kliinilist andmestikku, mis oli koostatud kardioloogiliste haiguste patsientide ravimise ja jälgimise käigus. Arendatud *baseline* rakendati nendele andmetele ilma ekspertteadmisteta.

Valitud meetodeid kasutati tulemuste genereerimiseks (klastrid, assotsiatsioonireeglid). Need tulemused edastati spetsialistile, et ta interpreteeriks ja hindaks neid domeeni kontekstis.

Lõpuks, valideeriti *baseline*'i empiirilist ajalist keerukust. See rakendati genereeritud mahukale andmehulgale, mõõdeti tööks kulunud aega.

Töö tulemusel rakendati väljatöötatud *baseline,* mis näitas, et admete analüüs on võimalik ilma domeeniteadmiseta. Teatud analüüsi etapid oleksid siiski effektiivsemad, kui eksperdi nõu oleks saadaval.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 70 leheküljel, 8 peatükki, 21 joonist, 15 tabelit.

List of abbreviations and terms

Data imputation	*Replacing missing values with substituted values*
Distance matrix	*Statistical distance (how different are objects) between data records in the view of the matrix*
FP	*Frequent Patterns*
P-value	*Statistical value, which shows the probability, that presented null-hypothesis is true*
RF	*Random Forest*
Similarity measure	*The measure used to calculate similarity between objects (distances)*
SMC	*Simple Matching Coefficient*
Shape of data	*Amount of columns and rows in dataset*
VAT	*Visual Assessment of Tendency*

Table of contents

List of figures

List of tables

1 Introduction

1.1 Background and motivation

Clinical data is complex in its origin. This kind of data often has different sources and is represented in different formats. For example, it could be collected automatically by performing certain procedures, or manually entered and provided by different laboratories or clinics. Those factors make data heterogeneous and voluminous [1]. Despite that, the complex clinical data is important for analysis, since it always contains useful and important information, located in the relationships of this data. Unfortunately, as the complexity of the medical data grows, it becomes difficult and time consuming to analyze it manually. Therefore, data analysis should be applied to extract useful knowledge from this data.

In general, data analysis is a complex process, focusing on extracting useful knowledge from a data. Usually, this process consists of following steps: data selection, data preprocessing, data transformation, data mining, interpretation and presentation [2]. Each of these steps is a separate collection of different techniques and methods, aimed to improve data analysis results. Those steps could be used in the iterative way, meaning, that during the whole process, some of the steps could be changed, adjusted and repeated in any point of time, depending on the results, which were provided by other steps. Data analysis is always very data-specific. Data properties such as volume, shape, type, variety, and overall quality have the crucial role in the data analysis and define the optimal flow of the process and the possible range of applicable techniques and methodologies.

By applying data analysis in the context of clinical data, data could be transformed into a form which would help to find valuable tendencies, dependencies and interrelationships within data. Pinpoint this information by carrying analysis manually is much more difficult and sometimes impossible.

1.2 Problem statement

The main aim of this work is to develop a data analysis baseline for sparse clinical data. This baseline will represent a clinical data analysis basis and will aimed to simplify and accelerate the analysis process. To work on this thesis, we were provided with real clinical data, which was gathered during the treatment and observation of patients with cardiac diseases and assembled, using trial form (Appx. 1). Provided data is a subset of bigger dataset and is represented in the view of the table, consisting of 150 records. Each record represents a patient's info and contain hundreds of properties (reaching 849). This represents another problem of clinical data analysis – data is represented in horizontal form. Having a large amount of properties, collected from different sources, assemble an appropriately large amount of records is usually difficult. This strongly affects data analysis accuracy and statistical expressiveness. Due to the complexness of provided data, manual analysis is impracticable.

It is noticeable, that the quality of collected data is extremely low. A lot of missing values, duplicates, misspellings and incorrect values are present. Besides, data is represented by multiple types, meaning that both categorical and continuous values are present. This imposes certain limitations in the data analysis process. This representation of data is not suitable for directly applying data mining techniques to get a meaningful result. Existing data should be processed and transformed in order to get an optimal result from data analysis.

Another problem of data analysis field is that domain knowledge, which is important in certain parts of data analysis, is not always directly accessible. It slows down the whole process and can lead to non-optimal or bias results.

Presented problems form the main complexity of this thesis. If not resolved, they might hinder gaining constructive knowledge from provided data, which will obstruct the successful knowledge discovery from provided data.

1.3 Goals

The main goal of this work is to develop data analysis baseline for sparse clinical data and apply it on provided dataset to produce an information, which could be useful for further analysis by domain expert. Considering the problems, mentioned above, it could be by divided into following steps:

1. Develop an appropriate data analysis baseline for sparse clinical data (see part *“2 Data analysis baseline structure”*).

2. Apply developed baseline to provided data and validate it, so that potentially useful information can be extracted and presented to domain experts for further study.

3. Develop an optimal baseline without knowledge and define whether presence or absence of relevant expertise affects data analysis procedures.

4. Provided data is an extracted subset from bigger dataset. Therefore, influence of dataset`s growth on computational time complexity should be analyzed.

2 Data analysis baseline structure

The standard baseline of data analysis consist of following steps (Fig. 1) [2]:

- Selection - extracting an appropriate data subset from the whole dataset.
- Preprocessing - processes aimed to improve the quality of data.
- Transformation - finding useful features of data and projecting it into a form, which will be optimal for application of data mining techniques.
- Data mining - applying techniques, such as clustering analysis, frequent pattern mining, etc. to find and produce valuable patterns and data structures.
- Interpretation – extracting an important information by understanding the produced patterns.

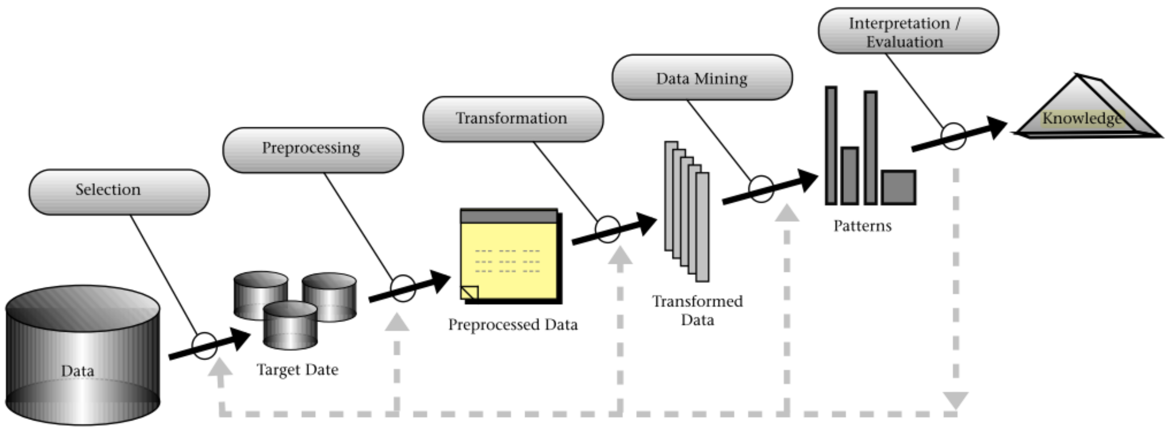


Figure 1: Standard baseline of data analysis [2]

In developed baseline preprocessing, transformation and data mining steps are more focused on.

The selection step is omitted, since dataset, which we were provided with, already contains the relevant data and this step always require expert knowledge.

Preprocessing step is represented by data cleaning techniques, including data transformation, missing data analysis and duplicates analysis.

Dimensionality reduction techniques were applied as transformation step.

Data mining includes association rules mining and clustering analysis (applying clustering techniques and validation of clustering techniques).

The interpretation step is also not a part of this thesis, because it requires deep domain knowledge. However, produced patterns will be provided to domain expert for further analysis.

The main idea of developed baseline is to take a sparse clinical data as an input and without applying domain knowledge, produce some patterns, which could be evaluated, interpreted and analyzed by domain expert.

3 Preprocessing

Clinical data is usually represented in high-dimensional form with the high amount of records. In most cases, data is highly heterogeneous. It is gathered from different sources, having different format and representation. Such data contain misspellings, inconsistencies, redundant and missing values, duplicates, etc. Certainly, it affects data quality which is one of the most significant aspects, necessary for successful extracting meaningful data analysis results.

Low-quality data complicates making significant and meaningful data analysis strategic decisions regarding building of the analysis baseline. To improve data quality, data cleaning could be applied.

3.1 Data cleaning

Data cleaning is a process of detecting incorrect, redundant and missing values and then correcting them. Being a complex of different techniques and approaches, it allows to improve the quality of data. In developed baseline was considered 3 data cleaning techniques – data transformation, missing data analysis and removal of duplicates. Implementation of those techniques is described in the *"7.1 Implementation of the processes"* part.

3.1.1 Data transformation

Clinical data is usually represented by multiple types (continuous, categorical). The type of data is one of the most limiting factors of data analysis. Some of the analysis techniques are only applicable to a certain data type to achieve an optimal result. To simplify the analysis, we can transform all data to one type.

Provided data is mostly categorical, but nonetheless properties with continuous data are also present. In order to find them and generalize the search of continuous data without applying domain knowledge, we can filter our data by finding columns with high proportion of amount of unique values.

Using this method, we have discovered following distribution:

- Categorical data: 811 columns (95.52%)

- Continuous data: 38 columns (4.48%)

Since most of our data is categorical, to simplify data analysis, we focus on categorical data only techniques. To apply them we would have to get rid of the continuous data first. However, since those columns could contain important information, their removal is undesirable solution. To keep this information, continuous columns could be converted to a categorical intervals. Normally, the most optimal way to implement this process would be to use the domain knowledge and apply data transformation, by creating categories based on this knowledge, which are more context-relevant. However, since domain knowledge was lacking in this study, categories from continuous columns could be also produced by replacing their values by number of uniform categorical intervals. For example, all column ‘age’ values, which vary from 33 to 88, could be replaced with 5 following categories: ‘33-44’, ’45-55’, ’56-66’, ’67-77’, ’78-88’. Such automatic transformation without domain knowledge could introduce a bias to the analysis, since categories, describing specific clinical information could be produced improperly. Nevertheless, we assume, that this approach would harm the analysis expressiveness less, than removing those columns altogether, due to uniform distribution of produced column values and the possibility to interpret the final results of analysis by domain expert.

3.1.2 Missing data analysis

The unavoidable property of most complex datasets is missing values [1]. This factor is a consequence of a whole range of reasons, such as input errors, difference in formats and data sources, context specifics. Missing values introduce decrease of analysis accuracy and lead to meaningless and misleading results. This is a fundamental factor in the definition of the data quality. To conduct a successful data analysis, missing data should be analyzed and properly handled.

3.1.2.1 Overview

In our data, there are 127350 values in general with 150 rows and 849 columns. By counting missing values, we discovered, that 76125 (59.78%) are missing. This is an extremely low index for a straightforward statistical analysis and definitely requires a processing. Currently, there is no strict and established threshold of the acceptable proportion of missing values [3]. But, obviously, reducing the amount of missing values would lead to an increase of data statistical expressiveness and analysis accuracy.

Curiously enough, along with fully filled columns, our data contain empty columns:

Number of columns	849
Completely filled columns	101 (11.9%)
Empty columns	170 (20.02%)

Table 1: Empty and completely filled columns amount

170 completely empty columns would introduce noise, misleading results and decrease analysis performance. Therefore, they were removed from the dataset. However, dataset is still sparse with high amount of missing values (Fig. 2)

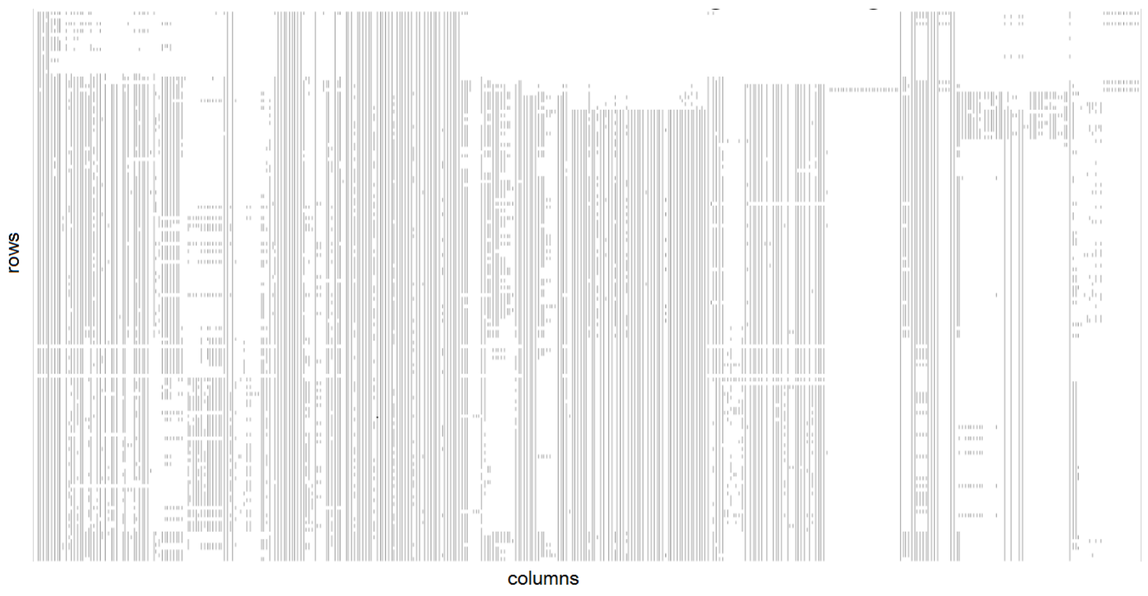


Figure 2: Visualization of dataset in the context of missing values before processing (white elements – missing values)

Statistical information presented in this thesis will not take into account completely empty columns. By calculating missing values by columns we have following distribution (Fig. 3).

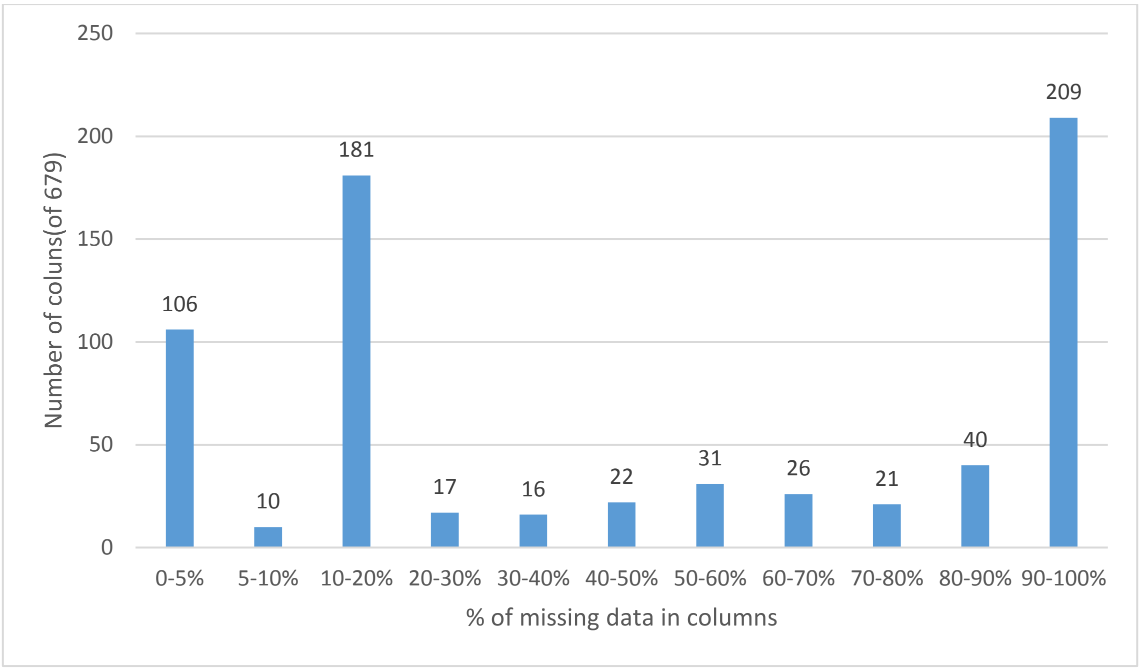


Figure 3: Missing data distribution by columns

Most of the columns are within 3 main categories: 0-20% missing (297 columns, 101 of them are completely filled), 90-100% missing (209 columns) and the uniform distribution with minor differences of columns within 20-90% (173 columns). This diverse distribution could decrease accuracy of results in regard to the whole data analysis process. Presence of such distribution complicates the search of dependencies between columns.

Besides understanding of missing data distribution between columns, it is important to analyze the distribution by rows (Table 2, Fig. 4).

Number of rows	150
Completely filled rows	0 (0%)
Partially filled	150 (100%)

Table 2: Empty and completely filled rows amount

Even with removing completely empty columns, the dataset have a missing values in every row.

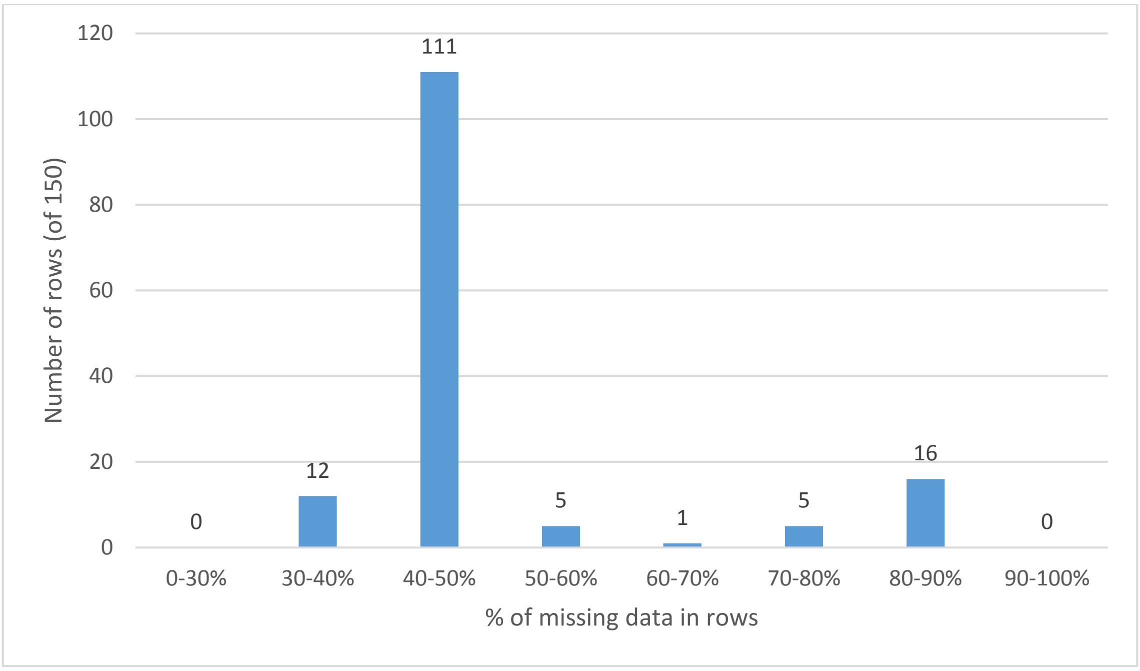


Figure 4: Missing data distribution by rows

This distribution shows, that most of the data rows are in the range of having 40-50% missing values.

High amount of the columns and rows with missing values might have a significant influence on data analysis results and make it impossible to use whole dataset in the following analysis steps. At this point the goal of data cleaning is to find possible ways to reduce amount of missing values, introducing minimal amount of bias to data expressiveness.

3.1.2.2 Missing data types

Missing values in the dataset could be represented by different types of the missingness mechanisms [4]. They describe the origin of the missing value and its significance in the dataset. It is important to determine them since different missingness mechanisms require different processing approaches. Wrong determination and processing of missingness mechanisms may lead to misinterpretation of the missing value meaning and could be a reason of distorted and biased results [5], [6].

Generally, there are 3 types of missingness mechanisms [4]:

- MCAR (missing completely at random) - There is no relationship between the missing value and other values, missing or observed. The occurrence of missing values is not systematic. In this case, missingness is completely independent from variable. For example, if the age of the patient was not reported due to some technical error.

- MAR (missing at random) – There is a systematic relationship between the missing value and the observed data, but not the missing data. Whether an observation is missing has nothing to do with the missing values, but with the values of others properties. For example, if patients with certain diseases tend not to fill up some fields in the form.
- MNAR (missing not at random) – There is a relationship between missing values and other values of property. For example, there could be a rule to not report some medical procedure results, if they are below some threshold.

When we are dealing with the MCAR data, the handling approaches could be more drastic, than with the MAR and MNAR data. For example, we can remove columns or rows with missing data without loss of valuable data. In the case of MAR and MNAR, the goal is to understand reasons of their appearance, interpret them and handle based on interpretation.

3.1.2.3 Handling missing data

There are several ways, how to determine mechanism of missingness. The most reliable is to apply domain knowledge and manually interpret the distribution of missing values. If domain knowledge is not accessible, the mechanism type can be determined based on data statistical properties.

For example, we can apply Littles MCAR test [10]. This test uses null-hypothesis, that all missing values are MCAR. To successfully reject this hypothesis the p-value of test should be less than 0.05. If Littles test produces p-value larger than 0.05, it could be considered as a weak evidence against null-hypothesis. We have applied Littles test to our dataset and received the p-value=0.674. This indicates that null-hypothesis could not be rejected and allows to assume that missing values in our data are of MCAR type. However, despite the fact that we cannot reject null-hypothesis, it is still possible, that our data contain different types of mechanisms. Determination of MAR and MNAR values is more complicated. It is not possible to define statistically whether missing values are of MAR or MNAR type. To define them accurately, they should be observed manually.

In our data we have noticed, that a lot of missing values are generated by columns with date values. We made an assumption, that some of them represent a date when a certain procedures were conducted. Such columns could be an example of MNAR values. Instead of removing those columns, their content could be replaced by Boolean values, so that the value would represent, whether a patient had a particular procedure.

Dataset contain 47 date-type columns with following missing values distribution (Table 3).

Missing data > 95%	15 columns
Missing data 90-95%	8 columns
Missing data 70-90%:	14 columns
Missing data less than 70%	10 columns

Table 3: Missing value distribution of date-type columns

Those columns was converted to Boolean type values.

In addition to missing value types, the efficiency of missing data handling techniques depend on another data properties. With the clinical data, the fact, that it often have small amount of records, imposes serious limitation on them. The most common methods, such as complete case analysis and imputation techniques [5] are often not applicable. For example, complete case analysis could not be used, due to small amount of records, since removing them we will lose an already a small number of rows, which define statistical expressiveness. Having very few records will also affect the results of imputation techniques. Those techniques allow to replace missing values by statistically appropriate ones. This, however, will introduce bias to the results, which grows proportionally to the decrease of the number of rows. Taking into account the properties of our dataset, it is more favorable to remove less important columns based on the number and type of missing values.

The columns with more than 20% of missing values were removed from dataset. As a result, considering, that we replaced date properties, 505 columns were removed. The remaining 344 columns now represent the completely different distribution of missing values, which visually is more dense and simple (Fig. 5).

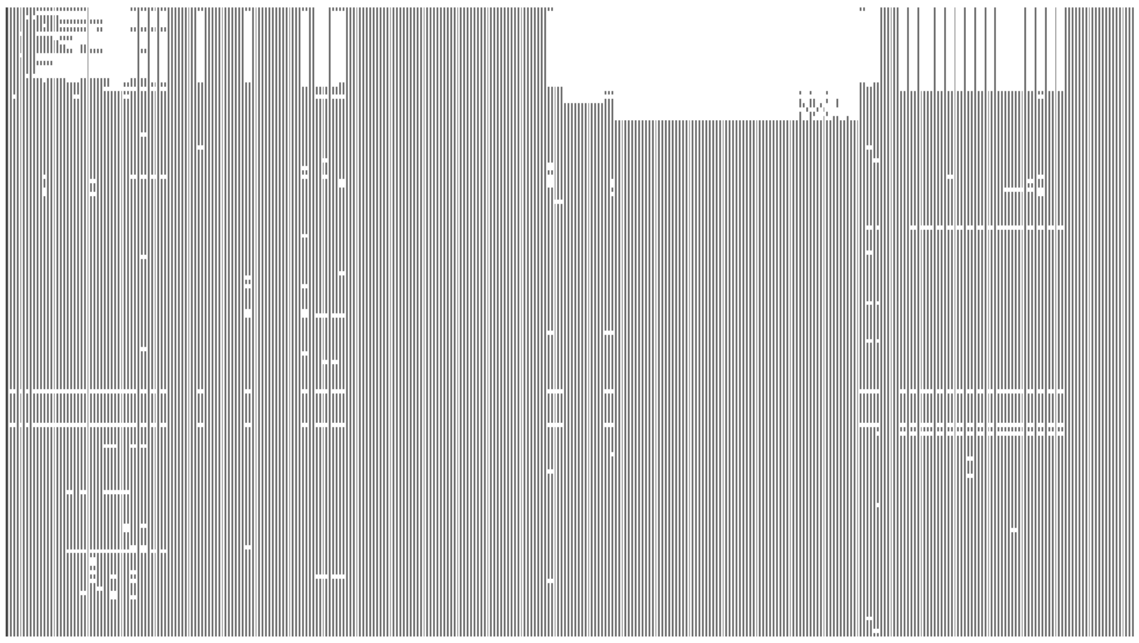


Figure 5: Visualization of dataset in the context of missing values after missing data handling (white elements – missing values)

The rows are now in the range of 0-5% of missing values (Fig. 6). Obviously, due to removing of columns, certain bias was introduced to dataset analysis. It is possible, that along with missing values, also meaningful values were removed from the dataset. Despite that, this data cleaning significantly changed representation of our data, by producing more restricted dataset with higher data quality (in the context of integrity of rows), making possible to continue data analysis.

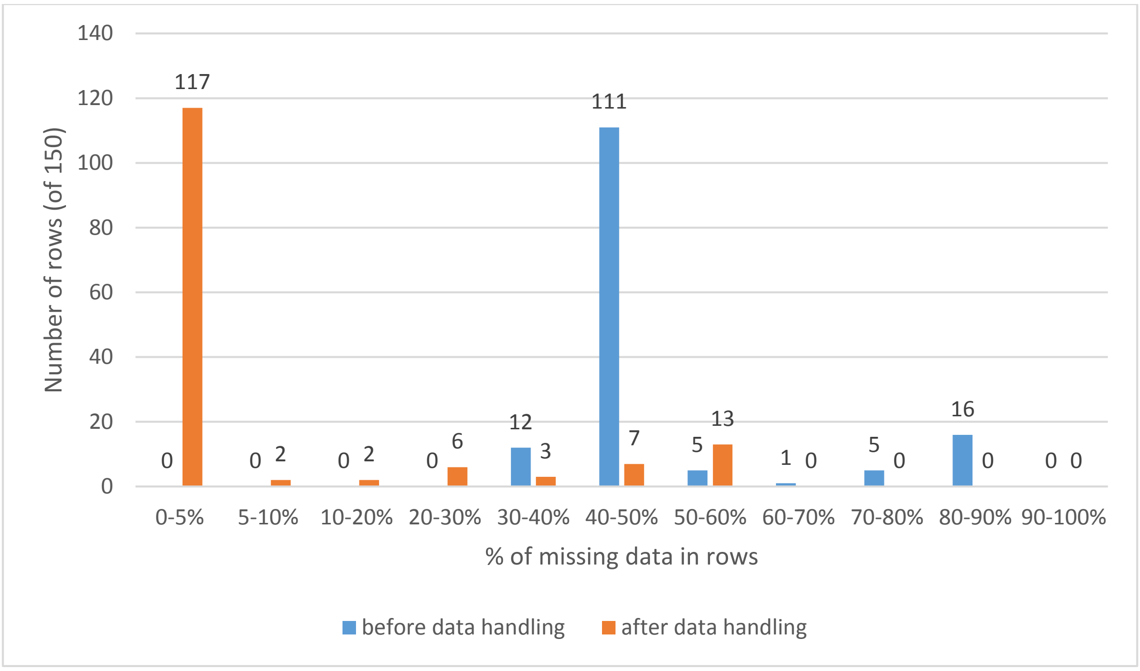


Figure 6: Distribution of missing values by rows before and after missing data handling

3.1.3 Removal of duplicates

Another problem of complex clinical data is duplicates. They introduce noise and redundancy to dataset and analysis. It is necessary to eliminate duplicated columns in order to bring consistency and improve the quality of the data [7]. Duplicates in our data are represented by 2 types - complete duplicates and context duplicates.

3.1.3.1 Complete duplicates

Complete duplicates represent completely identical columns and lead to inferior and misleading results. In provided data, complete duplicates was discovered by simply comparing columns with each other. Was found and removed 155 of 344 duplicate columns. This reduction will affect positively on following data mining methods in the context of performance and accuracy.

3.1.3.2 Context duplicates

In addition to complete duplicates, context duplicates could also exist in dataset. They represent the same property by multiple columns (e.g. patient sex is described by column as string “male/female” and by other column as code number “1/2”). Such duplicates are also redundant and should be removed from dataset.

Context duplicates could be identified by correlation analysis, where correlation represent a level of dependency between columns. To construct correlation matrix, we used chi-squared value [8], which is relevant to categorical data. Chi-squared test show the strength of the relationship between columns. By building the frequencies table [9] between all pairs of columns, it compares the expected frequencies of categories with observed frequencies of categories (Eq. 1). Based on this comparison, it calculates how strongly one column depends on another.

$$\mathcal{X}^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

Where:

$\mathcal{X}^2$ - Chi-squared test value

O_i - Observed frequency of type i

E_i - Expected frequency of type i

Equation 1: Chi-squared test equation

We can use this value, to calculate correlation coefficient between columns, by calculating the statistical effect of this value. For chi-squared test, it could be done by using Cramer's V coefficient calculation [10]. Using the chi-squared test value and shape of the data (Eq. 2), Cramer's V coefficient calculation produces a value between 0 and 1. This value also describes the strength of relationships between columns and could be interpreted, as correlation.

$$V = \sqrt{\frac{\mathcal{X}^2}{\min(n-1,\ m-1)}}$$

Where:

V - Cramer's V value

$\mathcal{X}^2$ - Chi-squared test value

n – Amount of rows in dataset

m – Amount of columns in dataset

Equation 2: Cramer's V equation

By calculating Cramer's V coefficient to all column pairs we constructed correlation matrix (Fig. 7). Column pairs with high Cramer's V value represent highly correlated columns. The pairs with Cramer's V coefficient that equals to 1 are context duplicates. Pairs with Cramer's V values close to 1 could also represent duplicates, where some of the values in columns contain misspellings. To extract duplicates we identified columns, which produce correlation coefficient more than 0.95.

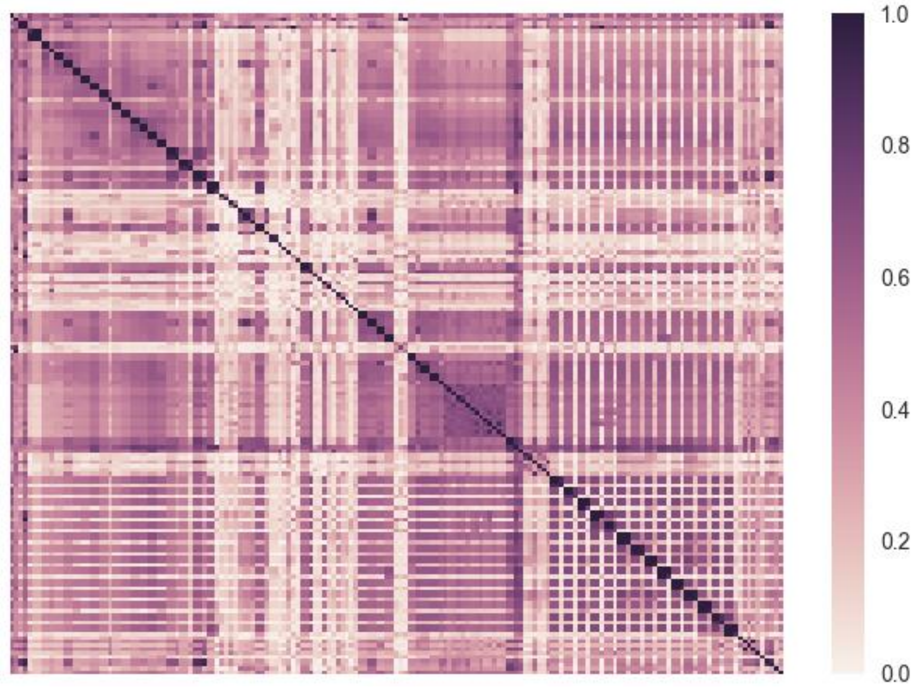


Figure 7: Correlation matrix of dataset without complete duplicates

In case of high-dimensionality dataset, calculating correlation using chi-square and Cramer's V is time consuming, since we have to build frequency tables and calculate correlation coefficient for all pairs of columns. By analyzing produced correlation matrix, we can notice, that highly correlated columns are near the main diagonal. This means, that those columns are located close to each other in dataset. To optimize duplicates reduction for further analysis with bigger datasets, we can divide dataset to smaller subsets (chunks) and analyzed them separately. This improves performance efficiency. Since most of duplicates are located closely to each other, this approach does not introduce high risk of skipping duplicates. To exclude this risk, we can apply context duplicates reduction on dataset several times by dividing dataset each time on chunks with different size. This approach eliminates risk of skipping duplicates and still will be more efficient, that analyzing whole dataset.

Our dataset with 189 columns (after removing complete duplicates) was divided into 7 chunks (27 columns each). For each chunk was conducted correlation calculation and columns with values more than 0.95 was extracted.

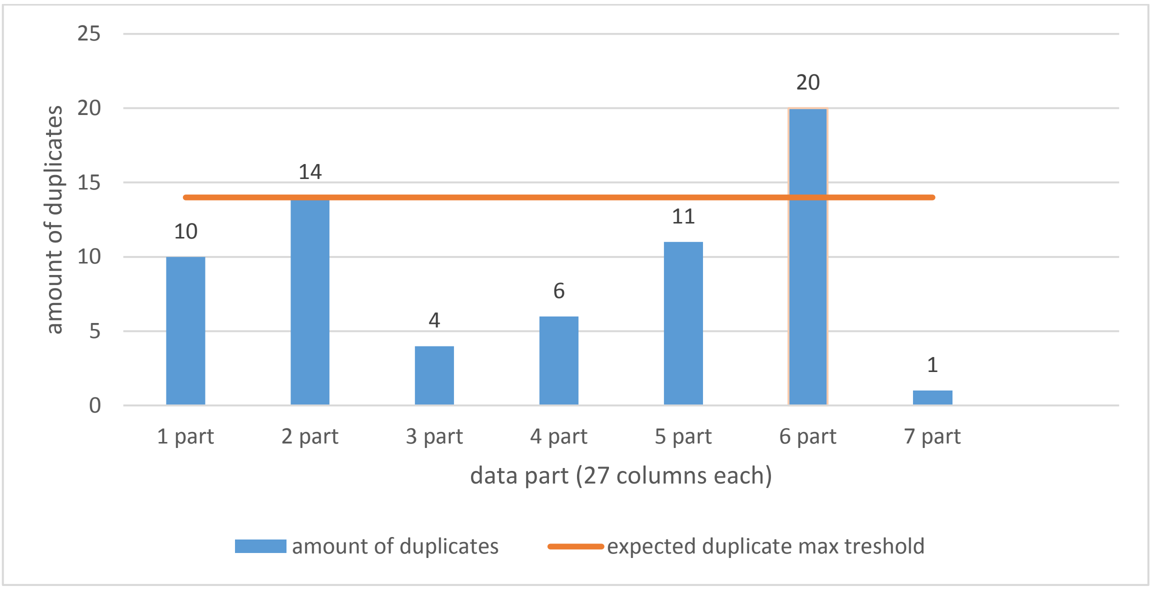


Figure 8: Distribution of context duplicates

66 context duplicates were discovered (Fig. 8). We expected, each chunk to have maximum 14 duplicate columns, meaning that each column in the chunk has a duplicate. In our result we got a chunk, which exceed this expected threshold. By analyzing result manually, we discovered, that often same property is described by 3 columns – name, code and date (which was transformed to Boolean). For example, columns such as VarasemadKardivaskulaarsedName, VarasemadKardivaskulaarsedKpv, VarasemadKardivaskulaarsedCode or vkhVarasemSkgKpv vkhVarasemSkgName, vkhVarasemSkgCode define same aspect of dataset. As a result, discovered duplicates were removed from dataset.

After removing complete and context duplicates data dimensionality decreased to 123 columns.

3.1.4 Data cleaning results

After applying missing data cleaning, the dimensionality of the data was significantly reduced without losing dataset records. Overall quality of the data was improved by reducing number of missing values and removing duplicates. Also, data was transformed to more convenient for analysis form.

2 datasets were produced for further analysis (Table 4).

Dataset	Columns	Description
1	679	Original dataset without completely empty columns.
2	123	Dataset with columns, containing 0-20% of missing values, without complete duplicates, without context duplicates.

Table 4: List of datasets after data cleaning

In further analysis original dataset without completely empty columns will be considered as *Dataset 1* and dataset produced by applying data cleaning as *Dataset 2*.

4 Transformation

Despite the fact, that by applying data cleaning, dimensionality of the dataset was significantly reduced, amount of dimensions could still be too big and insufficient for further data analysis [11]. In the context of clustering, high amount of dimensions increase the computational cost of clustering algorithms and makes the objective differences (distances) between data points appear less clear. This makes harder to understand relationships between records and form meaningful clusters. To reduce the amount of dimension, dimensionality reduction techniques [12] could be applied.

4.1 Dimensionality reduction

Dimensionality reduction will highlight the most and the least important features based on their statistical properties. We used those techniques, to generate extracted subsets additionally to datasets, which was produced by data cleaning (*Dataset 1* and *Dataset 2*).

4.1.1 Feature selection

One of the very efficient techniques of dimensionality reduction is feature selection [13]. It allows to extract a subset of features based on the statistical properties of the data. By highlighting more important features of the dataset it leaves up to the user to decide, which properties to extract. This allows for reduction of dataset by any number of dimensions.

In developed baseline we considered 3 feature selection methods:

- Feature selection by correlation
- Feature selection by random forest
- Feature selection by clustering validation.

Those methods was applied to *Dataset 2*, produced by data cleaning. Implementation of those methods is described in the *"7.1 Implementation of the processes"* part.

4.1.1.1 Feature selection by correlation

The most straightforward way to evaluate the importance of features is univariate feature selection, meaning, that statistical definition is calculated and evaluated for each feature individually.

One of the methods of univariate feature selection is to evaluate importance of features based on their correlation. For the implementation we can use correlation calculation solution, produced through early actions. The main this feature selection is to extract subsets of features, by removing highly correlated columns. By removing similar features in the context of correlation, the differences between data points will be more statistically expressive. The advantage of this method is, that on the application stage it not require domain knowledge. The columns, with correlation coefficient bigger than specified threshold were removed from dataset to produce a new subset. We applied this method to *Dataset 2* with different minimum correlation coefficient threshold (Table 5).

Minimum correlation coefficient threshold	Amount of dimensions in extracted subset
1.00	123
0.90	103
0.80	84
0.70	62
0.60	41
0.50	21

Table 5: Feature selection by correlation

To significantly reduce an amount of dimension in our dataset with this method, low coefficient threshold should be used (Table 5). Unfortunately, this introduces a risk to lose not only highly correlated redundant properties, but also relevant ones. As a result, we produced a subset of data with 21 columns (with correlation threshold 0.50). In further analysis it referred to as *Dataset 3*.

4.1.1.2 Feature selection by clustering validation

Another automatic method, which not requires domain knowledge is to extract important features by clustering application and evaluation. The idea is to conduct clustering on dataset with different amount of randomly extracted features and evaluate clustering accuracy. Then, extract subsets with highest evaluation results. This method require to conduct clustering analysis, therefore its implementation is presented in the *"5.1.3.3 Dimensionality reduction by clustering validation"* part.

4.1.1.3 Feature selection by random forest

Another approach for selecting features is to construct a random forest and evaluate the importance of features in this forest [14]. This approach is based on one predictor (represented by feature), which is used for building multiple decisions trees. By constructing such trees, mean Gini impurity value is calculated for each property. This value could be interpreted as an importance of the features for specified predictor.

This is a simple approach, which produces a straightforward overview of the importance of features. The approach is knowledge-based, since the predictor feature should be specified as an input. The appropriate choice of predictor should be done before application by domain expert. Therefore, we cannot include this approach in developed baseline. However, we will use it to evaluate an impact of dimensionality reduction on data analysis. Instead of selecting low-dimension data randomly, this approach could give a potentially more expressive subset.

We used feature "Surm sSurmaKuup2ev" as a predictor. By building forest with 100 decision trees (Appx. 2), mean decrease Gini value was calculated and 10 features with highest value was extracted (Fig. 9).

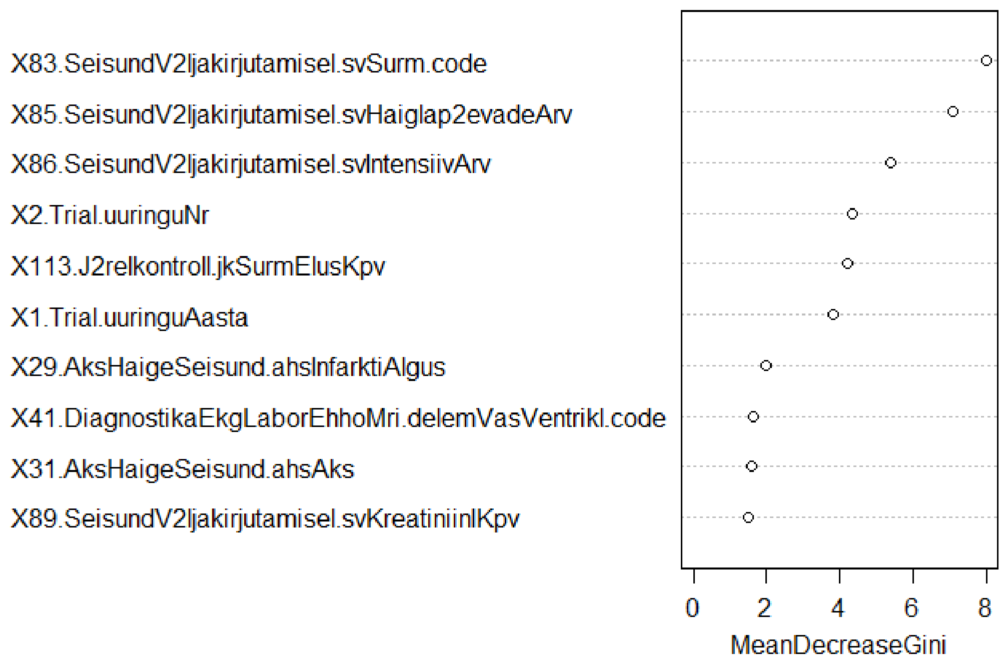


Figure 9: 10 most important features for "Surm aSurmaKuup2ev" predictor by random forest.

Those features were used to generate new subset (with 10 columns). In further analysis it referred to as *Dataset 4*.

5 Data mining

After data preprocessing and transformation, which produced 4 datasets, the next step of a developed baseline is to apply data mining techniques to extract patterns. We considered two techniques - clustering analysis and association rules mining. Implementation of those techniques is described in the *"7.1 Implementation of the processes"* part.

5.1 Clustering analysis

Clustering analysis is a process, which helps to understand the relationships between data points by grouping a set of records to clusters, based on their similarity. In developed baseline, clustering analysis consists of 3 steps:

- Clustering tendency - helps to understand, is the dataset useful for clustering.
- Clustering algorithms - divides data into clusters.
- Clustering validation - helps to understand the relevance of produced clusters and compare clustering algorithms.

5.1.1 Clustering tendency

Before applying clustering algorithms, we can validate our dataset on the presence of potential patterns. If data is two-dimensional, it could be done simply by visualizing the data. In the case of high-dimensional data, clustering tendency technique [15] could be used. This technique helps to understand, is the distribution within dataset uniform or it could contain possible patterns.

We implemented and included in the analysis baseline VAT [16] algorithm. This is clustering tendency approach, which is based on datasets ordered distance matrix (table of dissimilarity scores between all data points) visualization. It consists of following steps:

1. Distance matrix calculation - calculation of distances between all data points. Generated matrix will be symmetric with zeros on main diagonal.

2. Distance matrix order – order of distance matrix, so that distances ascend from main diagonal without violating the data records relationships. This means, that in ordering we can only operate with rows and columns but not directly with matrix cells. The matrix symmetry should be preserved.

3. Generation of random ordered distance matrix – creation of random dataset with the same shape as in original dataset and with respect to categories. This means, that variance of categories within generated dataset should be same as in original dataset. Using generated dataset, steps 1 and 2 should be repeated.

4. Visualization – visualization of ordered distance matrices of randomly generated and original datasets.

5. Visualization assessment – comparison of visualized produced in previous step.

To calculate distance matrix, we used simple matching measure [17]. This similarity measure is relevant to categorical data. To create a random copy of dataset, for each column we calculated amount of unique categories within this columns. Then, we randomly generated categories in the range of this amount and filled dataset with those categories. To order distances matrixes was implemented an algorithm, presented in this article [16] (see *"7.1 Implementation of the processes"* part). After that, we have visualized ordered distance matrices for randomly generated data (Fig. 10) and for *Dataset 2* (Fig. 11). Visualization of rest datasets could be found in the appendix (Appx. 3, Appx. 4, and Appx. 5).

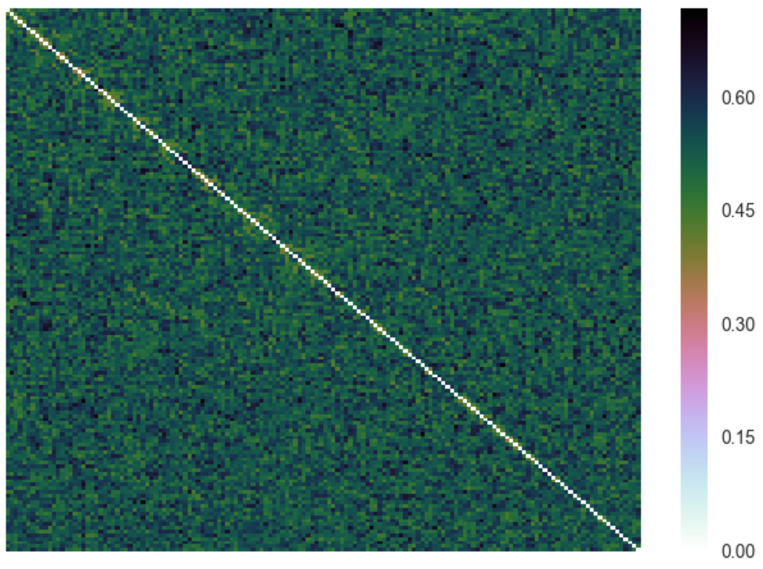


Figure 10: VAT for randomly generated ordered distance matrix (150 rows, 123 columns). Axises represent data points, color represent distance value.

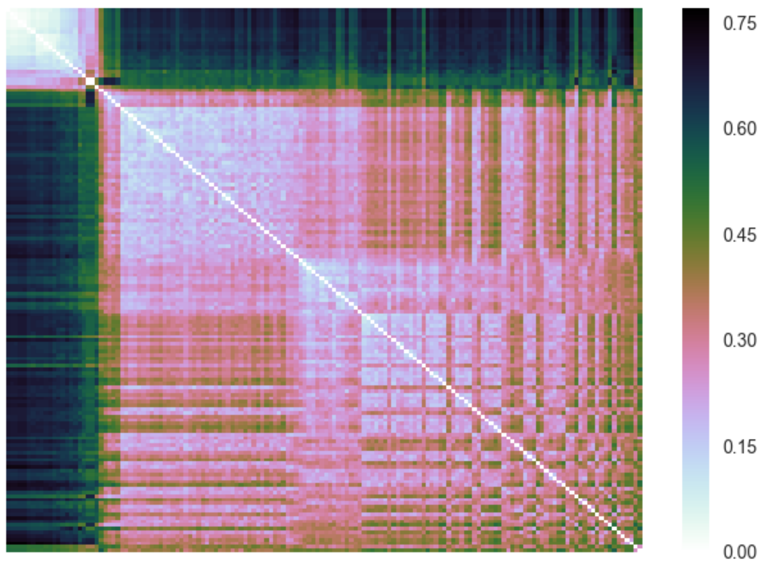


Figure 11: VAT for *Dataset 2* ordered distance matrix. Axises represent data points, color represent distance value.

The distribution of distance matrix based on random data clearly differs from matrix based on *Dataset 2*. When first represents a uniform distribution (Fig. 10), second has a potential patterns (Fig. 11). By comparing those figures we can assume, that *Dataset 2* could contain patterns and application of clustering algorithm would be relevant.

5.1.2 Clustering

If clustering tendency shows, that distribution of data is not uniform and could contain meaningful patterns, they should be discovered by applying clustering algorithms. The selection of the algorithms is not trivial. For different clinical datasets, optimal algorithms could be different. It strongly depends on the type of distribution within the dataset, which is difficult to identify before clustering. However, the selection could be done after clustering, by validating clustering results.

To make developed baseline more versatile, we applied two algorithms, which could handle datasets with different distribution types: centroid based k-modes [18] and density based HDBSCAN [19].

5.1.2.1 K-modes

Since we have discovered that our data could contain patterns, by applying simple matching distance measure, we used same distance measure to cluster data. Considering, that our data is categorical, we used k-modes algorithm [18]. K-modes is an extension of k-means algorithm. It replaces Euclidian distance measure (which is only applicable to a continuous data) by simple matching distance measure and identifies similar objects, by calculating modes instead of means. This extension preserves the efficiency of the k-means algorithm and makes possible to apply it on categorical data. K-modes algorithm is centroid based and is optimal for datasets with centroid-based distribution (Fig. 12).

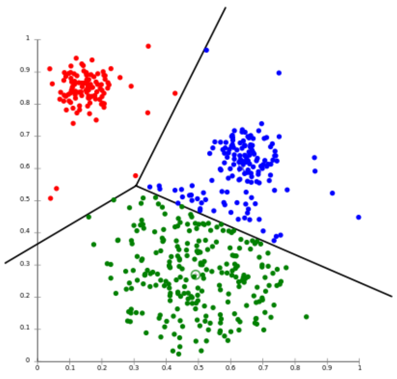


Figure 12: Centroid based distribution of data [20] (two-dimensional dataset)

One of the difficulties with k-type algorithms is that before clustering user should define the amount of clusters (*k*), to which the data will be divided. The optimal number of clusters depends on dataset structure and domain context. The only way to determine it before clustering is to use domain knowledge. However, to keep k-modes algorithm in developed data analysis baseline, we can determine optimal *k* after clustering. By applying algorithm multiple times with different *k* parameter, we can compare clustering results and choose the optimal input parameter (described in *"5.1.3.2.1 Selection of the optimal clustering parameters"* part).

K-modes algorithm was applied to datasets, produced in the previous steps (Table 6).

Dataset	Amount of columns	Amount of records in produced clusters, *k*=3	Amount of records in produced clusters, *k*=4	Mean computational time (seconds)
1	679	80/22/48	73/23/48/6	8.33
2	123	74/22/54	70/22/29/29	1.08
3	21	63/42/45	57/41/30/22	0.24
4	10	84/44/22	83/38/21/8	0.21

Table 6: Data records by clusters distribution using k-modes algorithm

After observation of produced clusters, we noticed, that results of the algorithm are very similar for *Dataset 1* and *Dataset 2*. They generate clusters with similar size, containing same objects. However, despite the fact, that data cleaning step did not affect the distribution of data in clusters, it introduces a significant improvement in algorithm performance in the context of time. The validation of k-modes results is described in *"5.1.3.2.2 Comparing clustering algorithms"* part.

5.1.2.2 HDBSCAN

Additionally to centroid-based distribution (Fig.12) data could be represented in more arbitrary forms (Fig. 13).

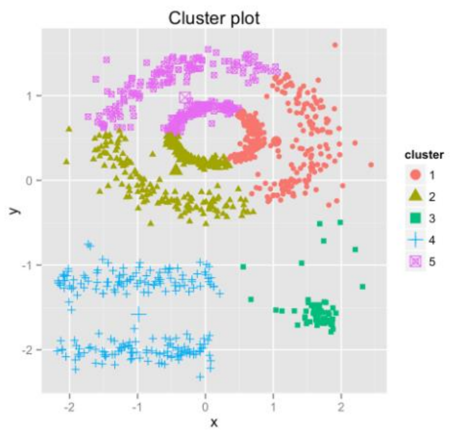


Figure 13: Arbitrary distribution of data [21] (two-dimensional dataset)

In this case application of centroid-based algorithms will be impractical. With such data, density-based algorithms should be used, which are very efficient in the case of arbitrary distribution. One of them is DBSCAN [22]. This algorithm operates with distance matrices and has no restrictions on the clustering similarity measure. With our dataset, we calculated similarity matrix using the same measure, as with k-modes algorithm. DBSCAN produces clusters by using 2 main input parameters:

1. *Eps* – radius of the neighborhood (restricted area around data record).
2. *MinPts* – minimum amount of points (data records) in the neighborhood.

Using those parameters, algorithm divides all data points into 3 groups:

1. Core - points, whose neighborhood within *Eps* radius contain *MinPts* or more neighbor points.
2. Border - points, whose neighborhood within *Eps* radius contain minimum 1 point and less, than *MinPts* neighbor points.
3. Noise - points, whose neighborhood within *Eps* radius does not contain any neighbor points.

With this separation, DBSCAN collects all core points, which are connected to each other to form clusters. Border points are used as the cluster borders. Noise points are considered as outliers and are ignored. This approach have following advantages:

1. Clusters of any shapes could be discovered.

2. Outlier points are ignored.
3. There is no need to determine the amount of clusters before clustering.

However, the main disadvantage of this algorithm is that produced result is very sensitive to the specified parameters. The determination of the optimal *Eps* and *MinPts* is not trivial and requires domain knowledge.

To include this algorithm into developed baseline and avoid using domain knowledge, HDBSCAN algorithm [19] could be applied. This is a hierarchical extension of DBSCAN with less sensitivity to input parameters. Basically, this algorithm performs DBSCAN over varying *Eps* parameter and extracts the clusters with best performance. This method allows returning an optimal result with the minimal tuning of the input parameters.

Unlike DBSCAN, this algorithm does not apply the same *Eps* neighborhood radius to all points. Instead, it iteratively increases *Eps* value and forms the density level for each data point. It creates a density tree (cluster hierarchy) by merging points to a cluster, when they occur within the same neighborhood (Fig. 14).

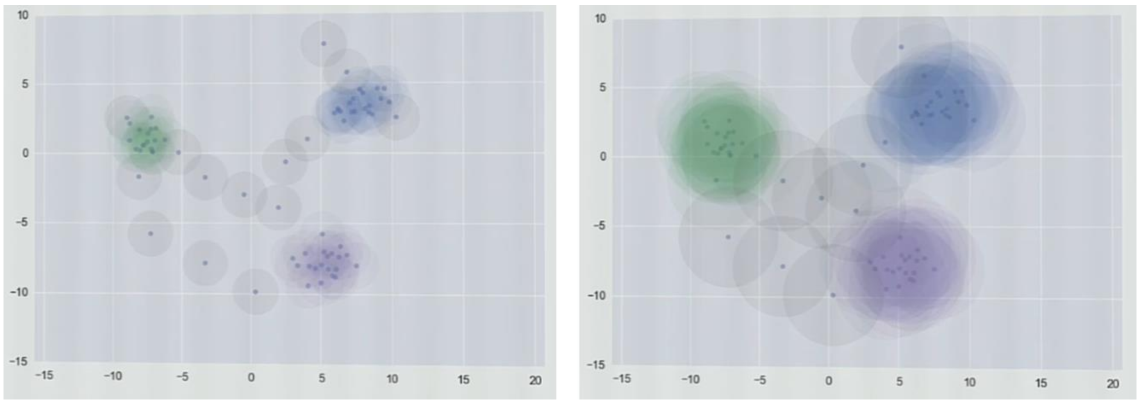


Figure 14: HDBSCAN forming clusters hierarchy by increasing *Eps* value (example for 2-dimensional data, where x and y axises represent feature values)

Since it splits the tree every time, when new point occurs in the neighborhood, tree quickly becomes too voluminous and unreadable. To bypass this problem, it uses minimum cluster size input parameter (amount of minimum objects in the cluster). This parameter defines a threshold, allowing to split density tree and simplifies produced tree. The optimal minimum cluster size parameter could be defined by clustering statistical validation. Therefore this algorithm could be included in the developed baseline.

HDBSCAN generates density tree in a form of the dendrogram (Fig. 15), where branches are potential clusters. Color and wideness of dendrogram branches represent the amount of data points in cluster and alpha value – how long cluster remained unchanged. Algorithm selects branches with longest alpha value as an optimal clusters (outlined with red, green and blue circles on Fig. 15). Children of selected branches are identified as noise values.

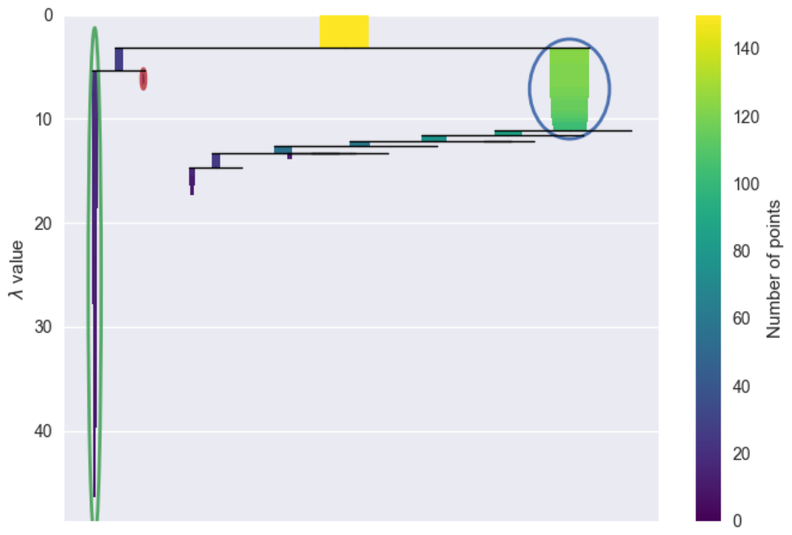


Figure 15: HDBSCAN generated density tree with Dataset 3, minimum cluster size = 5

We applied this algorithm to a produced before datasets, with minimum cluster size = 5 (Table 7).

Dataset	Amount of clusters	Amount of records in produced clusters	Amount of discovered noise points	Time complexity (seconds)
1	3	117/22/7	4	0.64
2	3	122/22/6	0	0.25
3	5	20/5/29/78/6	12	0.22
4	7	9/7/35/9/32/24/20	14	0.16

Table 7: Data records by clusters distribution using HDBSCAN algorithm

As with k-modes algorithm, *Dataset 1* and *Dataset 2* produced similar clusters. Clusters produced with *Dataset 3* and *Dataset 4* were very different. The validation of HDBSCAN results is described in *"5.1.3.2.2 Comparing clustering algorithms"* part.

5.1.3 Clustering validation

After applying clustering algorithms, the results should be analyzed. To understand clustering result, the validation techniques should be included in developed baseline. To give a completely objective evaluation and interpret clustering results in the domain context, domain knowledge is necessary. However, we can use clustering validation, avoiding domain knowledge to fulfill following goals:

- Evaluate the impact of the data cleaning on clustering results.
- Define optimal parameters for clustering.
- Compare the performance of HDBSCAN and K-modes clustering algorithms
- Select features with highest validation results

Clustering validation could be divided by 2 types:

1. Validation based on external criteria – evaluation of clustering results based on external knowledge about data, such as predefined distribution schemes, expected classification, etc.
2. Validation based on internal criteria - evaluating clustering results based on dataset properties without external information.

To apply validation without domain knowledge, we can only use internal criteria. In this work we used two validation techniques – visual validation and statistical validation. Also, clustering validation was used for dimensionality reduction as feature selection technique.

5.1.3.1 Visual validation

To give an initial assessment of clustering results we can use visualization. Since dimensionality of our data is high, we can't visualize data points directly. However, we can visualize distance matrices of datasets with the respect to produced clusters. We have ordered datasets in the same way, they are distributed by clusters. Then we recalculated the distance matrices for all datasets and visualized them (Fig. 16, Fig. 17). On this figures color represents a distances between data points. Visualization of such matrices could describe, how similar are data records within clusters and how different are clusters. The good clustering result should represent clearly isolated patterns with sharp borders.

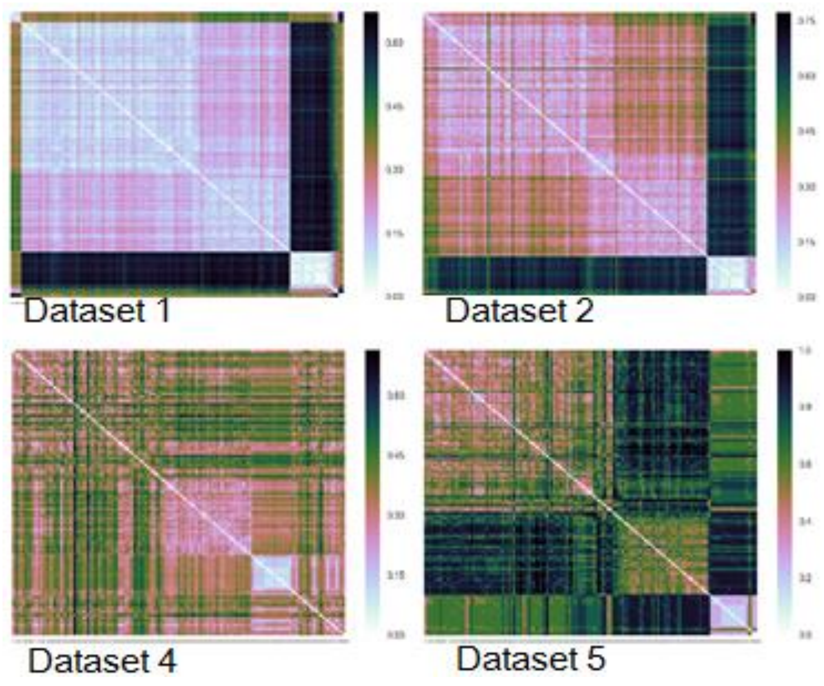


Figure 16: K-modes clusters visualization, k=3

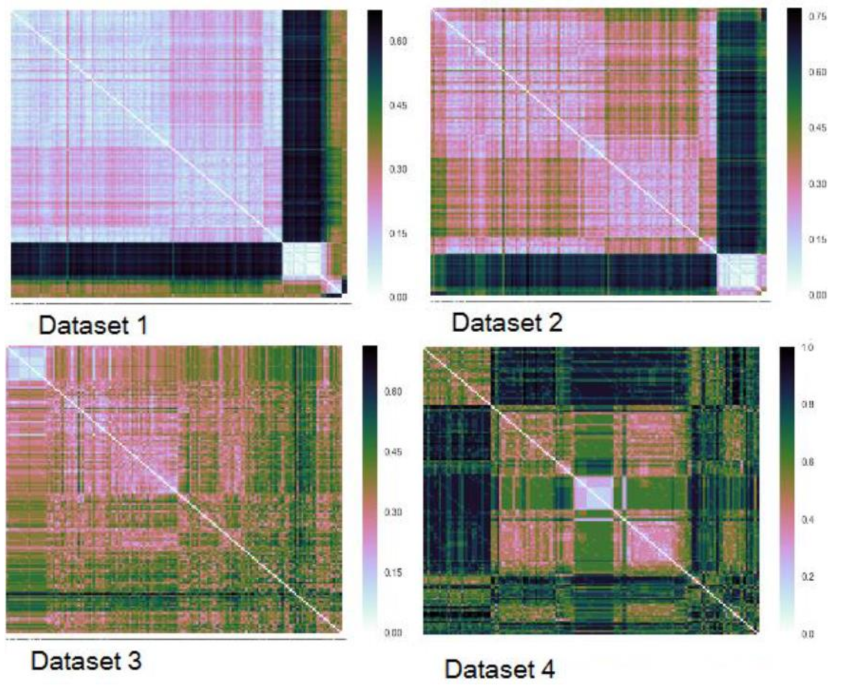


Figure 17: HDBSCAN clusters visualization, minimum cluster size = 5

The Fig. 16 and Fig. 17 represents k-modes and HDBSCAN clustering results visualization, for all produced before datasets. Both visualizations produce similar blocks (clusters) with *Datasets 1, 2* in the context of data distribution. However, clusters generated by *Dataset 2* have the most perceptible separation. This means that produced clusters are more distinct from each other. It indicates to more efficient clustering. Since *Dataset 2* represents a dataset, which was produced after data cleaning, we can assume, that data cleaning had a positive impact on clustering results.

Also, with both algorithms, the distribution of *Dataset 3* and *Dataset 4* is a lot sparser. It is more difficult to distinguish strongly outstanding patterns, which indicates poor clustering.

Having the poor efficiency of *Dataset 4* and *Dataset 5* with both clustering algorithm we assume, that feature selection by correlation with coefficient = 0.5 and by random forest is not efficient in the context of clustering. It does not improve clustering with our data.

Clustering visualization gave an initial overview of tendencies in produced clusters. However, since clustering was conducted with constant parameters, it does not provide an evaluation of optimal parameters. Also, we cannot compare performance of clustering algorithms with this approach. To receive this information, the statistical properties of produced clusters should be analyzed.

5.1.3.2 Statistical validation

In our case, statistical validation is based on internal indices. Currently existing a large number of various internal validity indices, which could be used for validation [23]. Generally, those indexes are based on two important clustering characteristics [24]:

1. Compactness - how similar are objects in cluster. It is based on within cluster distances, where lower distances indicate better compactness.

2. Separation - how distinct or well-separated are clusters from each other. Higher distances indicate better separation.

As a measure of compactness and separation, we use distances between data points in produced clusters. In this work as the internal index we applied Silhouette score [25], which shows a good accuracy with both centroid-based and density-based algorithms [24].

By calculating Silhouette score, we get a coefficient, which represents clustering efficiency, based on the pairwise distances between inner-cluster and intra-cluster data points. To calculate pairwise distances for our data, we used simple matching similarity measure.

Silhouette score varies between -1 and 1. High value indicates good clustering, where data points are well matched to clusters, to which they belong and poorly matched to other clusters. In developed baseline we use Silhouette score to define optimal clustering parameters, compare efficiency of clustering algorithms and select important data features.

5.1.3.2.1 Selection of the optimal clustering parameters

To define optimal parameters for clustering, we applied K-modes and HDBSCAN algorithms to all produced before datasets with different parameters (Fig. 18, Fig. 19).

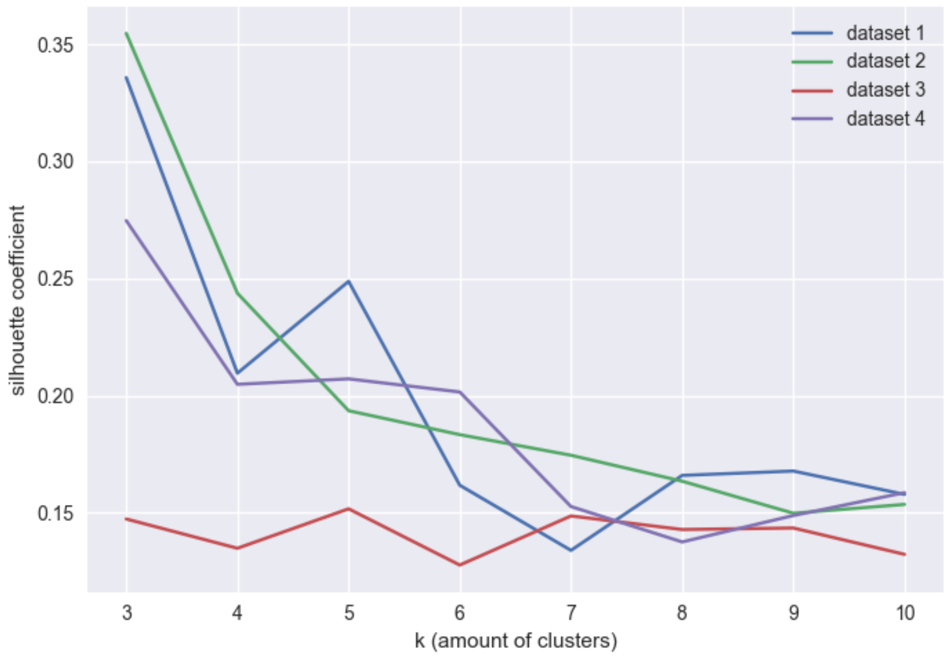


Figure 18: K-modes silhouette score for all data points with different amount of clusters

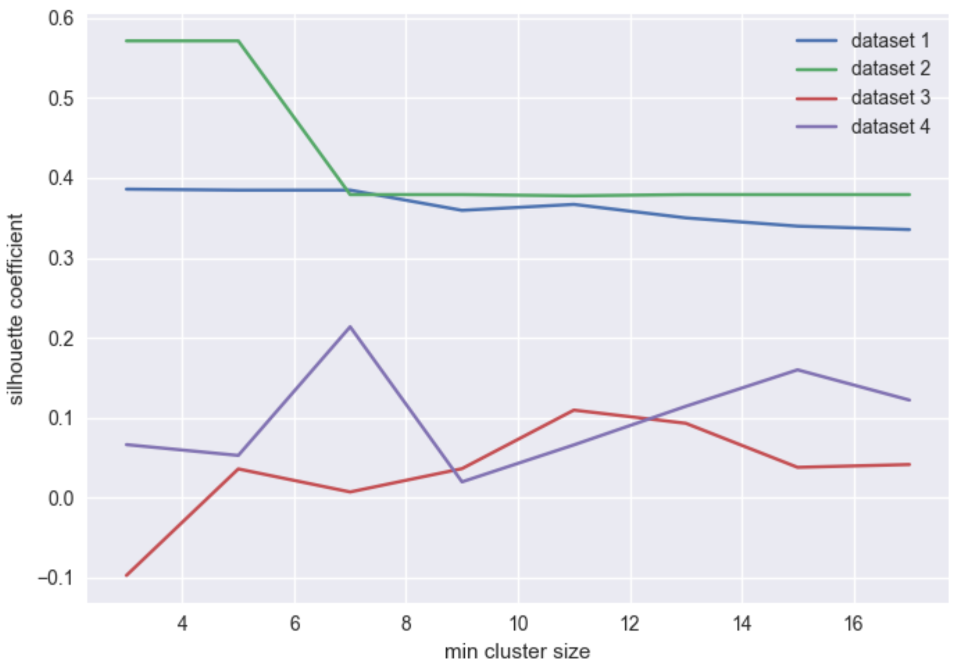


Figure 19: HDBSCAN silhouette score for all data points with different minimum cluster size

K-modes tends to have better clustering with almost all datasets, when data is separated by 3 clusters (Fig. 18). With increasing the *k* value, the quality of clustering decreases. The *k*=3 could be considered as the optimal parameter with clustering our dataset using k-modes algorithm. Also, it is noticeable, that worst results generate *Dataset 3* and *Dataset 4*.

In the case of HDBSCAN, we operate with minimum cluster size parameter (Fig. 19). Same as with k-modes, we have calculated silhouette score for all datasets with varying minimum cluster size and extracted the parameter, which generates highest validation score.

Clustering algorithms were applied to all datasets with optimal parameters. Produced clusters were sent to domain expert for evaluation and were used to compare the efficiency of clustering algorithm.

5.1.3.2.2 Comparing clustering algorithms

Additionally to finding optimal clustering parameters, we used silhouette score to evaluate the influence of data cleaning on clustering results and compare, which clustering algorithm is more relevant with our dataset. Clustering algorithms were applied to all datasets and silhouette score was calculated (Table 8).

K-modes algorithm produces highest silhouette score with *Dataset 2*. However, the difference between silhouette score with *Dataset 1* and *Dataset 2* is minimal. Significantly lower results were generated with *Dataset 3* and *Dataset 4*.

With the HDBSCAN, the best silhouette score also generates *Dataset 2*. However, unlike K-modes, the difference from other datasets was significantly bigger. This indicates on positive impact of data cleaning. As with K-modes, much worse performed *Dataset 3* and *Dataset 4.*

The best overall performance showed HDBSCAN algorithm. We assume, that HDBSCAN algorithm is more efficient to our dataset.

Dataset	Best K-modes silhouette score	Best HDBSCAN silhouette score
1	0.336	0.386
2	0.339	0.579
3	0.154	0.109
4	0.279	0.213

Table 8: Comparison of K-modes and HDBSCAN best silhouette scores with all datasets

By applying statistical validation, we confirmed assumptions made by initial visual validation. Application of data cleaning positively affects clustering results. However, the application of dimensionality reduction by correlation and random forest had a negative effect in the context of clustering. By using this technique in developed baseline, we can determine optimal clustering parameters and optimal algorithm for dataset, based on statistical parameters of the result.

5.1.3.3 Dimensionality reduction by clustering validation

Additionally, we used clustering validation as an approach of dimensionality reduction. At first we extracted multiple random subsets from our data with different amount of dimensions. Then we applied k-modes and HDBSCAN algorithms with those subsets and validated clustering results using silhouette score (Table 9). For every amount of dimensions, random subsets were picked 30 times.

Dimensions	3	5	10	20	50	90	120
K-modes silhouette score							
Min	0.089	0.107	0.152	0.208	0.281	0.266	0.207
Max	0.985	0.787	0.69	0.513	0.543	0.440	0.422
HDBSCAN silhouette score							
Min	0.453	-0.09	0.051	0.174	0.182	0.032	0.361
Max	0.994	0.993	0.921	0.827	0.815	0.773	0.771

Table 9: Min and max silhouette score distribution by applying k-modes and HDBSCAN to randomly extracted datasets

Some of data subsets, even with high amount of dimensions generated very accurate results in the context of silhouette score, especially with HDSCAN clustering. Subsets, which generate high silhouette score were extracted for further analysis. Also were extracted clusters, produced by those subsets, which was sent to domain expert for interpretation.

5.1.4 Clustering analysis results

In developed baseline clustering analysis is based on statistical properties of data and clustering results. However, statistical properties are not enough to interpret the result in the context of domain. For such interpretation, domain knowledge should be applied. Therefore, following information, produced with optimal parameters was sent to domain expert for the interpretation and further analysis:

K-modes:

- Clusters, produced with *Dataset 1*, *Dataset 2*, $k = 3$

HDBSCAN:

- Clusters, produced with *Dataset 1*, *Dataset 2,* minimal cluster size = 5
- Clusters, produced with subsets, selected by clustering validation based feature selection, with highest silhouette score (3, 5, 10, 20, 50 dimensions)

5.2 Association rules mining

Besides clustering, to extract additional information from clinical data, we applied association rules mining technique [26]. The main purpose of this technique is to fulfill following goals:

- Find frequent patterns in dataset
- Generate association rules, based on frequent patterns

Presence of those goals divide this technique by 2 steps:

1. Frequent pattern mining – process, aimed to find frequent patterns in the data, which pass defined minimum support threshold (varies from 0 to 100). This threshold indicates, how frequently those patterns appear in dataset.

2. Association rules generation – process, aimed to generate rules, based on frequent patterns, which pass defined minimum confidence threshold (varies from 0 to 100). Confidence is ratio between the rule and the rule basis appearing in the dataset.

This technique could be applied without domain knowledge and was included in the developed data analysis baseline.

5.2.1 Frequent pattern mining

Frequent pattern mining goes through dataset and finds the interesting patterns represented by data features called itemsets. Interestingness is measured by frequency of the pattern appearing in dataset. FP mining is computationally complex and presents a main difficulty in the association rules mining process. The computational complexity of FP mining significantly varies depending on dataset properties (shape of the data, variety of features) and on the mining algorithm. In this work, were considered 2 well-known algorithms - Apriori [27] and FP-Growth [28]. Considering advantages and disadvantages of both algorithms [29] and comparative study [30], we decided to use in our baseline FP-Growth algorithm, which shows much better performance, comparing to Apriori.

5.2.1.1 FP-Growth

FP-growth is an extension of Apriori algorithm, which removes its bottlenecks. It operates with dataset records to calculate frequencies of patterns and identify frequent patterns. When Apriori iteratively finds the frequencies for every pattern size, FP-Growth builds a tree with all frequencies only once. This approach gives significant increase of performance efficiency and reduces computational cost.

The main parameter of FP-Growth is minimum support. Patterns with low support are usually uninteresting in the context of analysis, since rules generated with them may occur simply by chance and not represent a meaningful association [26]. However, for the clinical data, there is no defined optimal value for minimum support, which will produce best results. Rules generated from patterns with various minimum support could interesting [31]. Considering this factor, we established the range of minimum support between 40 and 80. This range excludes patterns, produced by chance and allows to analyze rules with different support score.

One of the biggest problems of FP mining is that it tend to generate a big amount of patterns. Having high-dimensional data, by directly applying FP-Growth to our dataset (*Dataset 2*) and decreasing the minimum support, the amount of generated patterns grows very quickly (Fig. 20).

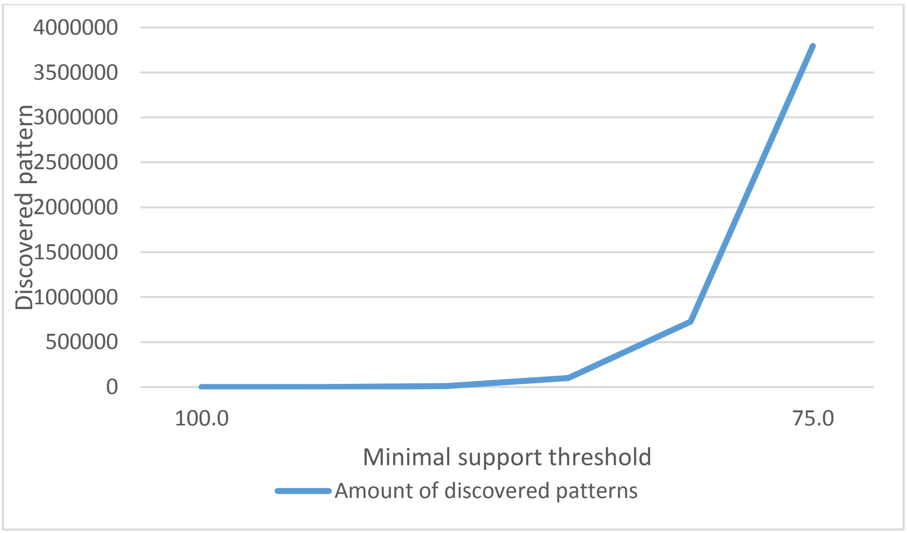


Figure 20: Growth of the amount of discovered patterns, using FP-Growth algorithm for *Dataset 2*.

With *Dataset 2*, amount of discovered patterns grows exponentially (Fig. 20). Already at support = 75 we get almost 4 000 000 patterns. It significantly reduces the effectiveness of analysis in the context of time complexity (Table 10) and in context of expressiveness of the results. It makes almost impossible to analyze patterns with high-dimensional data, introducing a huge amount of irrelevant and redundant information.

Minimum support	Time(seconds)
85	0.03
80	0.25
75	1.50
70	40.31
67	332.91

Table 10: Time complexity of FP-Growth algorithm, applied to *Dataset 2* with different minimum support parameters

To work around this problem, possible solutions could be considered from two different views:

1. From the view of the algorithm limitations
2. From the view of the dataset limitations

At first, from the view of algorithm limitations, we can reduce the maximum amount of items in discovered patterns. Frequent patterns will be used to generate rules, which will be sent to domain expert. This means, that eventually, they will be analyzed manually. According to studies [31], rules with more than 5 variables are hard to interpret in the context of clinical data. Considering this, we will limit the maximum amount of items per pattern by 5.

Results of this approach produce improvement in the context of time complexity (Table 11), allowing to generate patterns with lower support. However, growth of the number of patterns size is still exponential (Fig. 21).

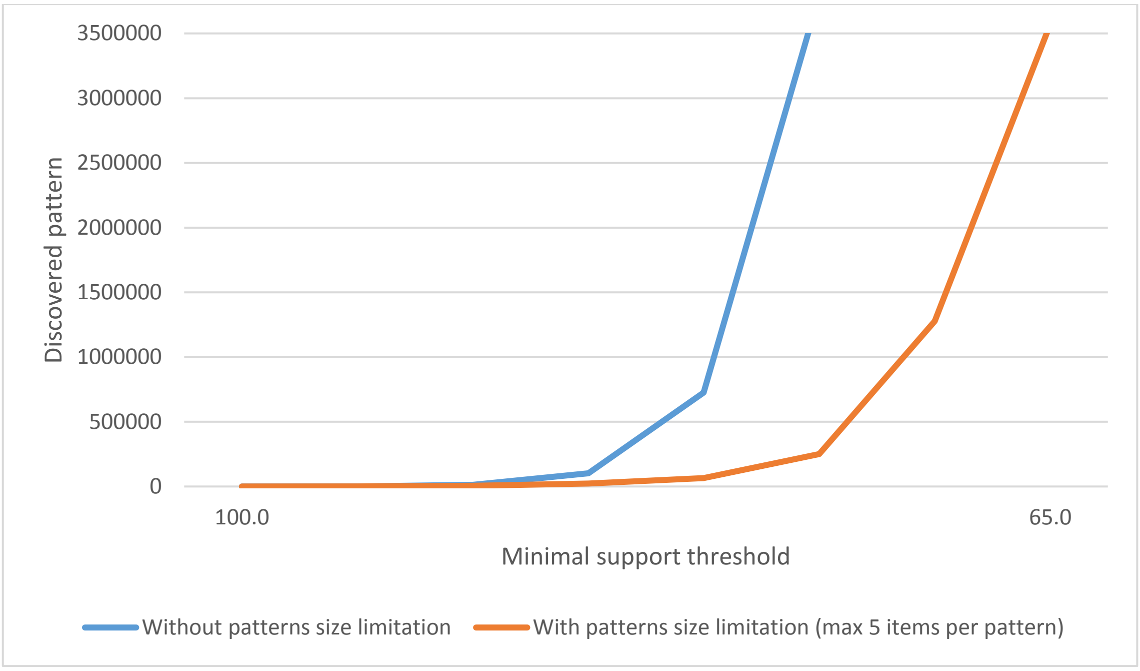


Figure 21: Growth of the amount of discovered patterns applying FP-Growth algorithm for *Dataset 2* with decreasing s, where s = minimal support, after applying pattern size limitation.

Minimum support	Time(seconds)
70	0.12
65	2.17
60	9.26
55	21.60
50	42.93

Table 11: Time complexity of FP-Growth applied to *Dataset 2* with pattern size limitation

From the view of the dataset, the biggest factor, affecting the number of generated patterns is the amount of dimensions. Since *Dataset 2* produce inappropriately big amount results, we applied frequent pattern mining to datasets, generated by dimensionality reduction (Table 12). Despite the fact, that datasets, generated by random forest and by correlation gave low results in the context of clustering, they could still contain the meaningful results in the context of association rules mining. Also was considered datasets, produced by dimensionality reduction based on clustering validation with 5, 10, 20 and 50 dimensions. In following analysis they are marked accordingly: *Dataset 5*, *Dataset 6*, *Dataset 7* and *Dataset 8*.

Dataset 3 (10 dimensions)			
Minimum Support	80	60	40
Patterns	0	0	11
Dataset 4 (21 dimensions)			
Minimum Support	80	60	40
Patterns	808	2432	4703
Dataset 5 (5 dimensions)			
Minimum Support	80	60	40
Patterns	17	20	26
Dataset 6 (10 dimensions)			
Minimum Support	80	60	40
Patterns	336	361	582
Dataset 7 (20 dimensions)			
Minimum Support	80	60	40
Patterns	59183	83330	125433
Dataset 8 (50 dimensions)			
Minimum Support	80	60	40
Patterns	176464	687969	1192298

Table 12: FP-mining results with different support parameter. *Datasets 3, 4, 5, 6, 7, 8*

By applying FP-growth algorithm with those datasets, we got results with significantly smaller amount of patterns. It became possible to extract patterns with whole established range of minimum support (40-80). Frequent patterns, produced by *Dataset 3*, *Dataset 5* and *Dataset 6* consist of appropriate number of patterns (for manual analysis) and could be used as a material for rules generation.

5.2.2 Generating rules from frequent patterns

After extracting frequent patterns from our datasets, we used them to generate association rules. This process is based on the confidence parameter. Confidence measures the reliability of the produced rules [26]. For the further analysis we extracted rules with high confidence threshold (Table 13). Generated rules for those datasets was extracted and sent to domain expert for further evaluation.

Dataset 3 (10 dimensions), confidence = 80			
Minimum Support	80	60	40
Rules	0	0	38
Dataset 5 (5 dimensions) , confidence = 80			
Minimum Support	80	60	40
Rules	55	55	55
Dataset 6 (10 dimensions), confidence = 80			
Minimum Support	80	60	40
Rules	1386	1456	2114

Table 13: Association rules mining. *Datasets 3, 5, 6*

Association rules mining is a final stage of developed baseline.

6 Complexity growth

Developed baseline was applied to extracted dataset with low amount of records. Due to which, the relevance of the baseline should be evaluated with bigger datasets. We synthetically generated 500 row, 1000 row and 5000 row datasets, applied baseline processes on them and measured the efficiency of developed data analysis baseline in the context of time complexity (Table 14). The number of dimensions was same as with cleaned data (123 dimensions).

Distance matrix calculation (123 dimensions)				
Rows	150	500	1000	5000
Exec. Time (seconds)	0.14	0.77	3.01	77.49
Correlation calculation, using Cramer’s V and chi-squared for duplicates removing and dimensionality reduction (123 dimensions)				
Rows	150	500	1000	5000
Exec. Time (seconds)	20.61	22.53	27.34	29.83
Random forest feature selection with 100 trees (123 dimensions)				
Rows	150	500	1000	5000
Exec. Time (seconds)	0.24	0.83	2.17	31.45
K-modes, *k*=3 (123 dimensions)				
Rows	150	500	1000	5000
Exec. Time (seconds)	0.99	5.57	9.64	29.93
HDBSCAN without distance matrix calculation, min cluster size = 5 (123 dimensions)				
Rows	150	500	1000	5000

Exec. Time (seconds)	0.01	0.04	0.13	3.27
Silhouette score calculation without distance matrix calculation (123 dimensions)				
Rows	150	500	1000	5000
Exec. Time (seconds)	0.01	0.02	0.09	2.64

Table 14: Time complexity benchmarks with generated datasets

Frequently pattern mining and association rule mining depends mostly on data distribution within datasets. In randomly generated dataset with minimum support more than 1 it is hard to find a lot of patterns, due to dataset uniform distribution. However, by applying those algorithms for different datasets with minimum support more than 1, there was no difference in the context of computational time.

After applying benchmarks, the sharpest growth was with simple matching distance matrix calculation. However, it could be an implementation issue, since k-modes algorithm, using same dissimilarity measure performed much more efficient. Despite that, overall time complexity with bigger datasets is within reasonable time limits (in worst case scenario baseline application does not take more than 10 minutes). This means that developed baseline is relevant with the bigger datasets.

7 Implementation

Most of the techniques, used in this thesis were implemented with Python and R languages and open source libraries, as a project “medcl”. For more convenient work with datasets, almost all implementations used “pandas” and “numpy” libraries.

Source code and data could be found here: *https://github.com/jjjmm/medcl*

General project structure:

- *src – project source code*
- *data – directory with used and produced datasets, plots, baseline results*

Most of the baseline processes was implemented as separate files.

7.1 Implementation of the processes

Data transformation:

- src/continuous_to_categorical.py (python)
  - Description: Detection of continuous variables and conversion of those variables to uniform categorical intervals

Missing data analysis:

- src/data_stats/missing_data/missing_visualization.py (python)
  - Description: Dataset visualization in the context of missing data
  - Used libraries: “missingo”
- src/data_stats/missing_data/missing_data.py (python)
  - Description: Calculation of the distribution of missing values in dataset by rows and columns; removal of columns with high amount of missing values

Duplicates removing:

- src/duplicates.py (python)
    - Description: identifying duplicate columns and removing them
- src/cramers_v.py (python)
    - Description: Calculation of Cramer’s V score, correlation matrix visualization.
    - Used libraries: “matplotlib”, “scipy”, ”seaborn”.

Feature selection:

- src/feature_selection /rand_forest.r (R)
    - Description: Features selection by random forest generation
    - Used libraries: “randomForest”
- src/feature_selection/by_clust_validation.py (python)
    - Description: Feature selection by clustering validation
- src/feature_selection/by_corr.py (python)
    - Description: Feature selection by correlation

Clustering tendency:

- src/data_stats/tendency.py (python)
    - Description: implementation of VAT ordering algorithm, clustering tendency visualization
    - Used libraries: “seaborn”, “matplotlib”

Simple matching similarity measure:

- src/validation/simple_matching.py (python)

    - Description: Distance matrix generation based on simple matching similarity measure

K-modes clustering algorithm:

- src/k_modes.py (python)
    - Description: Clustering, using k-modes algorithm
    - Used libraries: “kmodes”

HDBSCAN clustering algorithm:

- src/hdbscan_impl.py (python)
    - Description: Clustering, using HDBSCAN algorithm, visualization of HDBSCAN hierarchy
    - Used libraries: “hdbscan”, “seaborn”, “matplotlib”

Clustering validation:

- src/validation/va.py (python)
    - Description: Implementation of visual clustering validation
    - Used libraries: matplotlib, seaborn
- src/validation/silhouette.py (python)
    - Description: calculating silhouette score for clustering validation and dimensionality reduction by clustering validation
    - Used libraries: “sklearn”, “matplotlib”, “hdbscan”

Dataset generation:

- src/util/data_util (python)
    - Description: Generation of synthetic datasets for time complexity evaluation

Littles test:

- External software - IBM SPS Statistics 23 (https://www.ibm.com/us-en/marketplace/spss-statistics)
    - Description: Calculation of Little’s MCAR test

FP-growth feature selection, Extracting rules from data mining:

- External software – FP-Growth algorithm implementation (http://borgelt.net/doc/fpgrowth/fpgrowth.html)
    - Generation of frequent patterns and association rules

7.2 Used datasets

Datasets, which were used in this thesis and were produced by application of developed baseline are listed in Table 15.

Path	Description
data/datasets/original.csv	Original dataset
data/datasets/generated/123_500.csv	Generated dataset with 500 rows
data/datasets/generated/123_1000.csv	Generated dataset with 1000 rows
data/datasets/generated/123_5000.csv	Generated dataset with 5000 rows
data/datasets/1_679.csv	Original dataset without completely empty columns.
data/datasets/2_123.csv	Dataset with columns, containing 0-20% of missing values, without complete duplicates, without context duplicates.

data/datasets/3_21.csv	Dataset, produced by correlation based feature selection
data/datasets/4_10.csv	Dataset, produced by random forest feature selection
data/datasets/5_3.csv	Dataset, produced by clustering validation feature selection (3 columns)
data/datasets/6_5.csv	Dataset, produced by clustering validation feature selection (5 columns)
data/datasets/7_10.csv	Dataset, produced by clustering validation feature selection (10 columns)
data/datasets/8_20.csv	Dataset, produced by clustering validation feature selection (20 columns)
data/datasets/randomly_filled_with_respect_of_categories.csv	Randomly generated dataset with respect of categories (used in clustering tendency assessment)

Table 15: Used datasets

8 Summary

During this work, data analysis baseline for clinical dataset was developed. This baseline consists of preprocessing, transformation and data mining techniques and could be applied without domain knowledge. Developed baseline could be used as an initial analysis base and accelerate whole data analysis process. Since data analysis is an iterative process, developed baseline could be changed and improved on any stage, relying on the interpretation of produced results.

Some of the techniques were implemented and validated using simple matching similarity measure with categorical data. However, since most of the techniques operate with distance matrices, they could be applied to other data types, using different similarity measure.

By applying data cleaning techniques, sparse data was projected to a more expressive form. This was the reason of achieving better results in the context of clustering. By using dimensionality reduction it became possible to extract association rules from dataset. Association rules mining and clustering analysis produced patterns, which could potentially represent valuable information. Developed baseline consists of following steps:

1. Data cleaning
    a. Data transformation
    b. Missing data analysis
    c. Duplicates removing
2. Dimensionality reduction
    a. Feature selection by correlation analysis
    b. Feature selection by clustering validation
3. Clustering analysis
    a. Clustering tendency

    b. K-modes
    c. HDBSCAN
    d. Clustering validation
4. Association rules mining
    a. FP-growth feature selection
    b. Extracting rules from frequent patterns

Developed data analysis baseline was applied to a real clinical dataset. With our data, HDBSCAN showed better results, than k-modes. Application of feature selection by correlation analysis and random forest produced a poor efficiency in the context of clustering. However, k-modes and feature selection by correlation remained in the baseline, since using those with clinical datasets with different data distribution could give better results. Feature selection by random forest was removed from baseline, since it could not be applied without domain knowledge.

During this work the developed data analysis baseline was created taking into account the absence of domain knowledge. However, decisions made in this work were based only on statistical properties, which always introduce some bias. Using domain knowledge with some techniques (data transformation, missingness mechanism determination) could improve final results. Also domain knowledge is necessary to interpret the final results and get an objective evaluation of baseline performance.

One of the biggest drawbacks of developed baseline is that strategic decisions were made using dataset with low amount of records. This factor reduces expressiveness of data and could cause biased results. Produced baseline should be validated by applying it with bigger real clinical dataset. However, even with small dataset, this baseline produces results, which could be potentially valuable in the domain context. Those results were sent to domain expert for evaluation and interpretation.

Also, since we worked with extracted subset, the performance of developed baseline was analyzed in the case of the bigger dataset. If the dataset will grow, it will not affect the performance significantly, therefore baseline could be used with bigger datasets.

References

[1] K. J. Cios and G. W. Moore, “Uniqueness of medical data mining,” *William Moore / Artif. Intell. Med. 26*, vol. 26, pp. 1–24, 2002.

[2] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From Data Mining to Knowledge Discovery in Databases,” *AI Mag.*, vol. 17, no. 3, p. 37, 1996.

[3] Y. Dong and C.-Y. J. Peng, “Principled missing data methods for researchers.,” *Springerplus*, vol. 2, no. 1, p. 222, 2013.

[4] “How to Diagnose the Missing Data Mechanism.” [Online]. Available: http://www.theanalysisfactor.com/missing-data-mechanism/. [Accessed: 19-Apr-2017].

[5] D. Schmidt, M. Niemann, and G. Lindemann-Von Trzebiatowski, “The handling of missing values in medical domains with respect to pattern mining algorithms,” *CEUR Workshop Proc.*, vol. 1492, pp. 147–154, 2015.

[6] “Missing Data: Solutions and examples (MCAR, MAR, MNAR).” [Online]. Available: http://www.odmguide.com/data-analysis-2/missing-data-solutions-and-examples/. [Accessed: 19-Apr-2017].

[7] B. Khan, A. Rauf, H. Javed, and S. Khusro, “Removing Fully and Partially Duplicated Records through K-Means Clustering,” *IACSIT Int. J. Eng. Technol.*, vol. 4, no. 6, 2012.

[8] M. L. McHugh, “The chi-square test of independence.,” *Biochem. medica*, vol. 23, no. 2, pp. 143–9, 2013.

[9] “Contingency Tables.” [Online]. Available: http://onlinestatbook.com/2/chi_square/contingency.html. [Accessed: 20-Apr-2017].

[10] “Measures of Nominal Level Association.” [Online]. Available: http://www.people.vcu.edu/~pdattalo/702SuppRead/MeasAssoc/NominalAssoc.html. [Accessed: 20-Apr-2017].

[11] “The Curse of Dimensionality in Classification.” [Online]. Available: http://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification#The_curse_of_dimensionality_and_overfitting. [Accessed: 03-May-2017].

[12] L. J. P. Van Der Maaten, E. O. Postma, and H. J. Van Den Herik, “Dimensionality Reduction: A Comparative Review,” *J. Mach. Learn. Res.*, vol. 10, pp. 1–41, 2009.

[13] T. H. Cheng, C. P. Wei, and V. S. Tseng, “Feature selection for medical data mining: Comparisons of expert judgment and automatic approaches,” *Proc. - IEEE Symp. Comput. Med. Syst.*, vol. 2006, pp. 165–170, 2006.

[14] K. J. Archer and R. V. Kimes, “Empirical characterization of random forest variable importance measures,” *Comput. Stat. Data Anal.*, vol. 52, no. 4, pp. 2249–2260, 2008.

[15] “Assessing clustering tendency: A vital issue - Unsupervised Machine Learning - Easy Guides - Wiki - STHDA.” [Online]. Available: http://www.sthda.com/english/wiki/assessing-clustering-tendency-a-vital-issue-unsupervised-machine-learning#hopkins-statistic. [Accessed: 20-Apr-2017].

[16] J. C. Bezdek and R. J. Hathaway, “VAT: a tool for visual assessment of (cluster) tendency,” *Proc. 2002 Int. Jt. Conf. Neural Networks. IJCNN’02 (Cat. No.02CH37290)*, vol. 3, pp. 2225–2230, 2002.

[17] “Simple matching coefficient.” [Online]. Available: https://en.wikipedia.org/wiki/Simple_matching_coefficient. [Accessed: 03-May-2017].

[18] Z. Huang, “Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values,” *Data Min. Knowl. Discov.*, vol. 2, no. 3, pp. 283–304, 1998.

[19] L. McInnes, J. Healy, and S. Astels, “hdbscan: Hierarchical density based clustering,” *J. Open Source Softw.*, vol. 2, no. 11, pp. 1–3, 2017.

[20] “434px-KMeans-Gaussian-data.svg.png (434×467).” [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/thumb/e/e5/KMeans-Gaussian-data.svg/434px-KMeans-Gaussian-data.svg.png. [Accessed: 04-May-2017].

[21] “DBSCAN: density-based clustering for discovering clusters in large datasets with noise - Unsupervised Machine Learning - Easy Guides - Wiki - STHDA.” [Online]. Available: http://www.sthda.com/english/wiki/dbscan-density-based-clustering-for-discovering-clusters-in-large-datasets-with-noise-unsupervised-machine-learning. [Accessed: 04-May-2017].

[22] M. Ester, H. Kriegel, J. Sander, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.”

[23] B. Desgraupes, “Clustering Indices,” no. April, 2013.

[24] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, “Understanding of intenal clustering validation measures,” *IEEE Internatinal Conf. Data Min.*, pp. 911–916, 2010.

[25] P. J. Rousseeuw, “Silhouettes - a graphical aid to the interpretation and validation of cluster analysis. Rousseeuw 1987,” vol. 20, pp. 53–65, 1987.

[26] P.-N. Tan, M. Steinbach, and V. Kumar, “Association Analysis: Basic Concepts and Algorithms,” *Introd. to Data Min.*, pp. 327–414, 2005.

[27] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Assoc. Rules,” 1995.

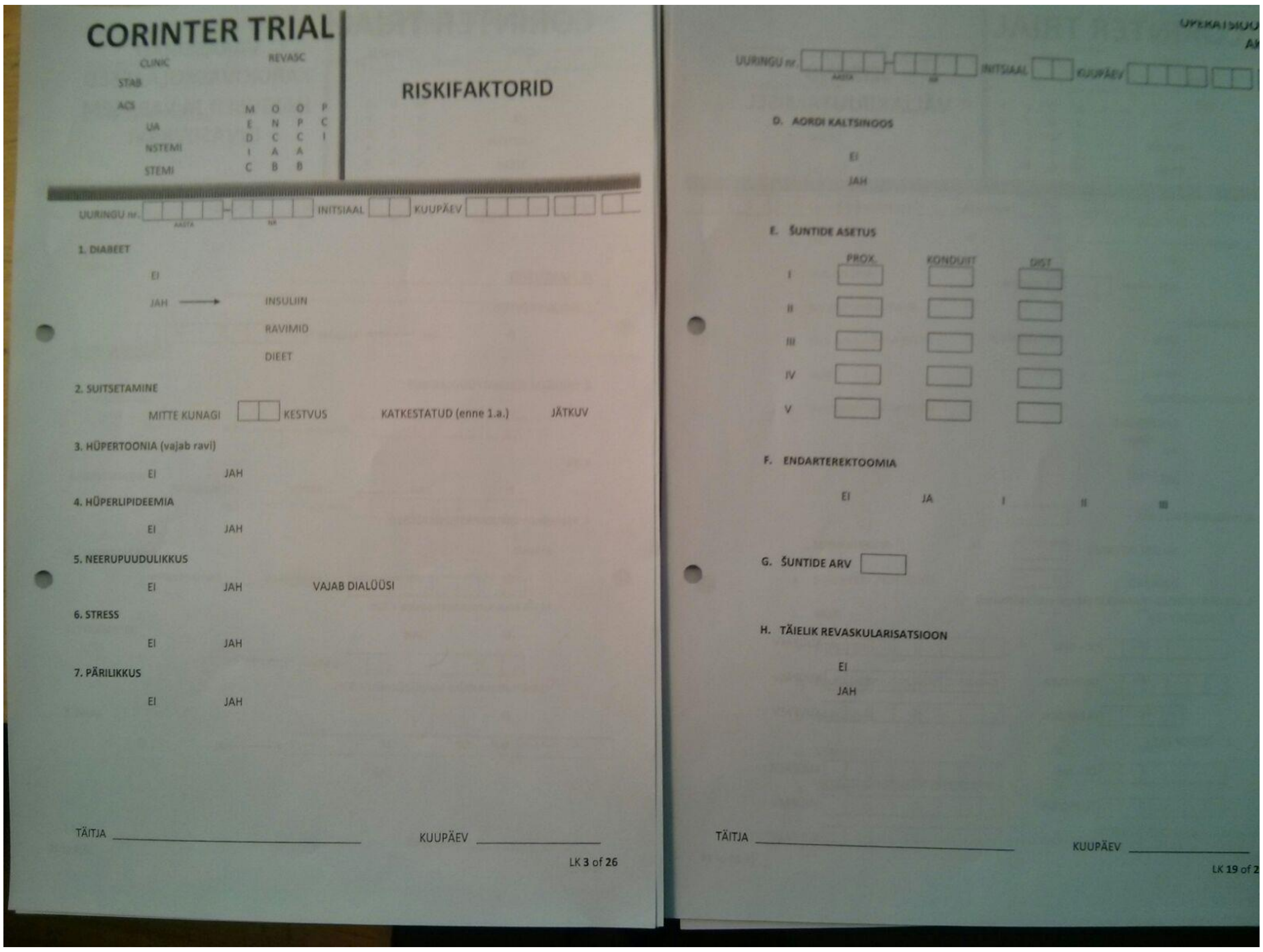
[28] J. Han, J. Pei, and Y. Yin, “Mining Frequent Patterns without Candidate Generation,” *Networks*, pp. 1–12, 2000.

[29] “Apriori vs FP-Growth for Frequent Item Set Mining” [Online]. Available: http://singularities.com/blog/2015/08/apriori-vs-fpgrowth-for-frequent-item-set-mining. [Accessed: 06-May-2017].

[30] K. Garg, “Comparing the Performance of Frequent Pattern Mining Algorithms” vol. 69, no. 25, pp. 29–32, 2013.

[31] C. Ordonez and C. A. Santana, “Discovering Interesting Assoc. Rules in Medical Data,” 2000.

Appendix



CORINTER TRIAL

CLINIC

STAB

ACS

UA

NSTEMI

STEMI

REVASC

M	O	O	P
E	N	P	C
D	C	C	I
I	A	A	
C	B	B	

RISKIFAKTORID

UURINGU nr. AASTA – NR INITSIAAL KUUPÄEV

1. DIABEET

EI

JAH → INSULIIN

RAVIMID

DIEET

2. SUITSETAMINE

MITTE KUNAGI KESTVUS KATKESTATUD (enne 1.a.) JÄTKUV

3. HÜPERTOONIA (vajab ravi)

EI JAH

4. HÜPERLIPIDEEMIA

EI JAH

5. NEERUPUUDULIKKUS

EI JAH VAJAB DIALÜÜSI

6. STRESS

EI JAH

7. PÄRILIKKUS

EI JAH

TÄITJA KUUPÄEV

LK 3 of 26

OPERATSIOO

AK

UURINGU nr. AASTA – NR INITSIAAL KUUPÄEV

D. AORDI KALTSINOOS

EI

JAH

E. ŠUNTIDE ASETUS

	PROX.	KONDUIT	DIST
I			
II			
III			
IV			
V			

F. ENDARTEREKTOOMIA

EI JA I II III

G. ŠUNTIDE ARV

H. TÄIELIK REVASKULARISATSIOON

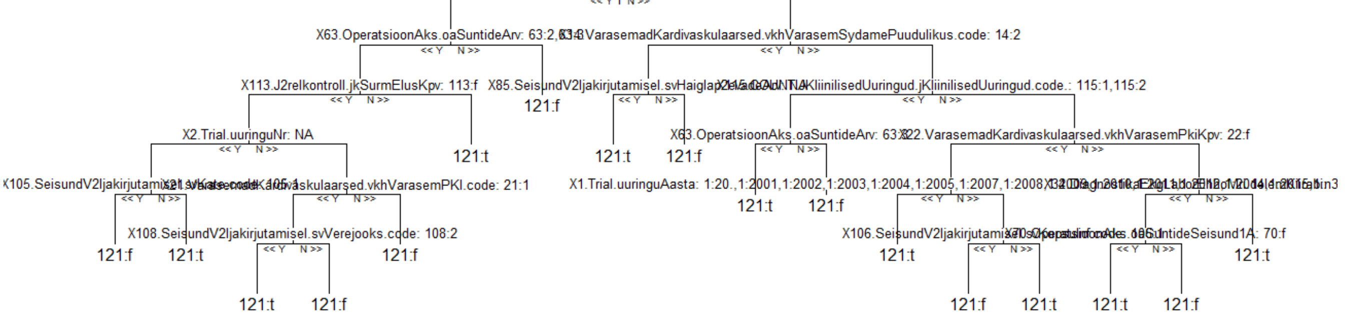
EI

JAH

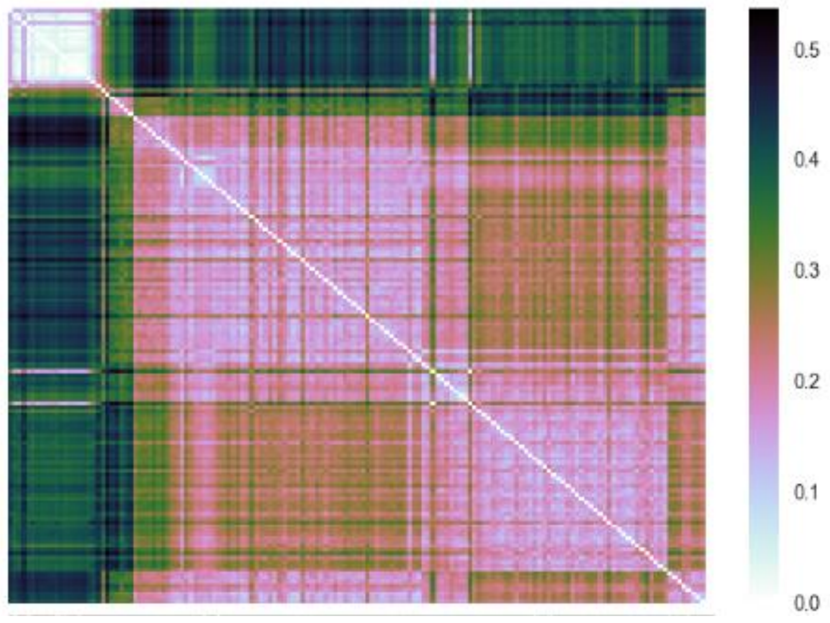
TÄITJA KUUPÄEV

LK 19 of 2

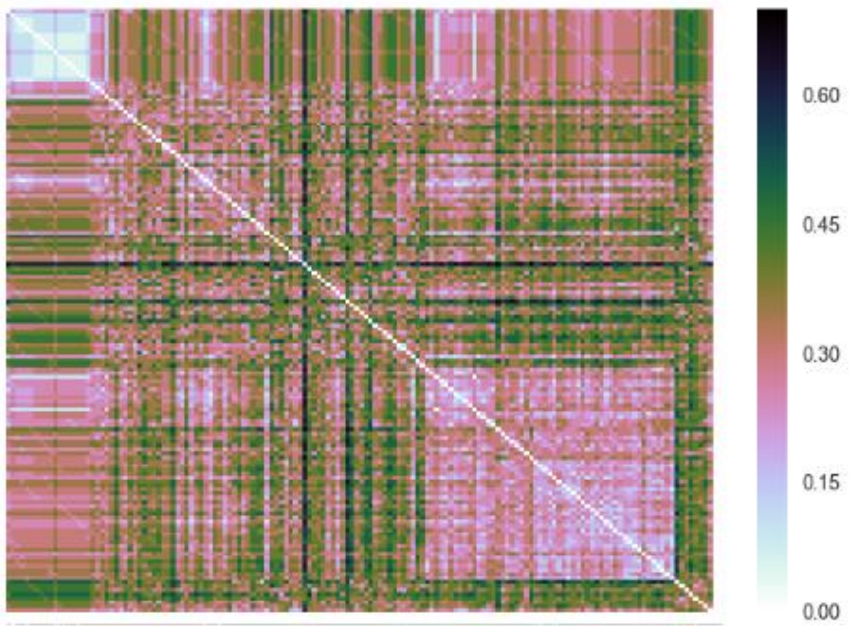
Appendix 1: Part of the form used to collect clinical data


X30.AksHaigeSeisund.ahsHospitaliseerimine: 30:f
<< Y N >>
X63.OperatsioonAks.oaSuntideArv: 63:2,63:3
X14.VarasemadKardivaskulaarsed.vkhVarasemSydamePuudulikus.code: 14:2
X113.J2relkontroll.jkSurmElusKpv: 113:f
X85.SeisundV2ljakirjutamisel.svHaiglapäevadeArv: NA
X115.KliinilisedUuringud.jKliinilisedUuringud.code.: 115:1,115:2
121:f
X2.Trial.uuringuNr: NA
121:t
121:t
121:f
X63.OperatsioonAks.oaSuntideArv: 63:3
X22.VarasemadKardivaskulaarsed.vkhVarasemPkiKpv: 22:f
X105.SeisundV2ljakirjutamisel.svKates.code: 105:1
X21.VarasemadKardivaskulaarsed.vkhVarasemPKI.code: 21:1
X1.Trial.uuringuAasta: 1:20.,1:2001,1:2002,1:2003,1:2004,1:2005,1:2007,1:2008,1:2009,1:2010,1:2011,1:2012,1:2014,1:2015
121:t
121:f
X108.SeisundV2ljakirjutamisel.svVerejooks.code: 108:2
X106.SeisundV2ljakirjutamisel
X70.OperatsioonAks.oaSuntideSeisund1A: 70:f
121:f
121:t
121:f
121:t
121:t
121:t
121:f
121:f
121:t
121:t
121:f

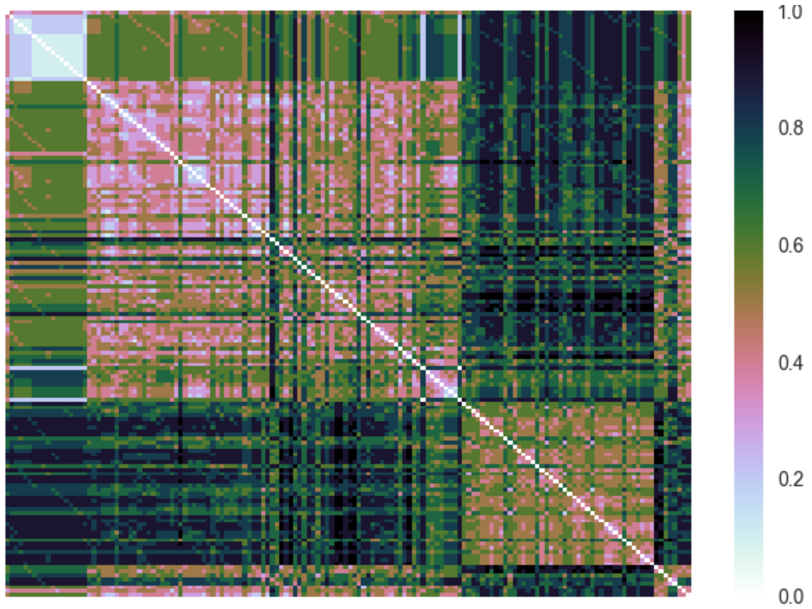
Appendix 2: Example of the decision tree in Random Forest



Appendix 3: VAT for *Dataset 1* ordered distance matrix



Appendix 4: VAT for *Dataset 3* ordered distance matrix



Appendix 5: VAT for *Dataset 4* ordered distance matrix