UKRAINIAN CATHOLIC UNIVERSITY

BACHELOR THESIS

Clustering online-marketplace customers' heterogeneous data applying unsupervised learning methods

Author:
Kostiantyn HRYTSIUK

Supervisor:
Mykola BABIAK

A thesis submitted in fulfillment of the requirements for the degree of Bachelor of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2022

Declaration of Authorship

I, Kostiantyn HRYTSIUK, declare that this thesis titled "Clustering online-marketplace customers' heterogeneous data applying unsupervised learning methods" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Viam supervadet vadens"

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Bachelor of Science

Clustering online-marketplace customers' heterogeneous data applying unsupervised learning methods

by Kostiantyn HRYTSIUK

Abstract

The common challenge in understanding the insights from the online customers' data is their segmentation. Since they have both continuous and categorical features, there is no straightforward way to obtain valuable clusters built on features of different types.

In this work, we want to compare existing algorithms for clustering mixed data and the application of different methods to measure non-euclidian distances between data points. The effectiveness of each "algorithm - distance measure" pair will be evaluated on the real-life subscription customers dataset.

Acknowledgements

I want to express my sincere gratitude to my parents and family for all kindness and patience that they gave me.

Also, I want to thank Borys Gudziak for all the faith and efforts to create Ukrainian Catholic University as it is now.

I am thankful to all team of Applied science faculty for such incredible community that can be safelly named as a family.

Big thanks to my supervisor Mykola Babiak for guiding me during the process of this research.

And especially, I am grateful to Oksana for her support and for going with me through this way.

Contents

List of Figures

List of Tables

List of Abbreviations

LCM	**L**atent **C**lass **M**odel
LCA	**L**atent **C**lass **A**nalysis
PAM	**P**artitioning **A**round **M**edoids
SOM	**S**elf- **O**rganized **M**ap

To all those whose victims will never be known and whose contribution will never be fully appreciated

Chapter 1

Introduction

The problem of grouping big sets of objects into segments that differs by some characteristics is common in many fields. Finding such relations between many objects to extract features prevailing for one group and not for another is a complicated and multilateral process. It becomes especially difficult when one deals with both quantitative and qualitative features of one object. If it is possible to calculate the distance in different ways for numerical features, it is not possible for categorical ones.

For example, we can say that one is heavier than the other among two apples by comparing their weights. However, we cannot do the same to compare their colors (omitting the comparison of those colors' hex values). Furthermore, suppose we still want to simultaneously group apples based on their weight and color. In that case, we will be dealing with heterogeneous data (weight and color) of each data point (each apple in population).

This example can be extrapolated to any other area, but we will be focusing on the online-marketplace business in this paper. The specific of this field is the vast number of customers visiting such businesses every day to fulfill their needs. The business stakeholders may want to understand better who those customers are and what they have in common.

The clustering methods of unsupervised learning can solve such a problem. In the context of clustering the heterogeneous data, the Jain&Dubes formulated this problem as follows:

"Clustering is an unsupervised machine learning technique used to group unlabeled data into clusters that contain data points that are 'similar' to each other and 'dissimilar' from those in other clusters" [8]

In this work, we will research existing solutions for clustering mixed data to define methods that are the most appropriate to the problem defined above. It will be done by covering existing scientific literature related to this topic. Based on outcomes from this process, we will select the set of methods to compare their effectiveness in clustering by applying them to real-world data. For that, we will be using data from the online-marketplace JustAnswer.

We will discuss the characteristics of clusters obtained from applying the best methods and provide recommendations regarding which clustering methods can be used as an effective tool for customer segmentation.

Chapter 2

Related Works

Clustering the mixed data is a broad scientific topic under study for more than half a century. Many papers cover different methods of solving this problem and applying them to real-world problems. In the first part of this chapter, we will go through the works that focus on the general overview of existing solutions. In the second part, we will provide a detailed look at the literature with a description of the application of clustering techniques for the mixed data from business areas related to the research object.

2.1 Literature review of existing clustering algorithms

The need to divide the real-world data sets that consist of both quantitative and qualitative variables into valuable sub-parts is typical across different domains. Moreover, such popularity resulted in various algorithms for solving this problem. Ahmad and Khan, in their work "Survey of State-of-the-Art Mixed Data Clustering Algorithms" [1], introduced the taxonomy of those methods that consist of 5 big cohorts.

2.1.1 Partitional methods

The main idea of this class of methods consists of three parts:

- **Center of each cluster** that combines both continuous and categorical features;
- **A measure of distance** between two observations that includes all types of their features;
- **Objective loss function** that minimizes during algorithm

Pros:

- Linearity of an algorithm;
- Simple scalability for large datasets;
- Potential adoption of parallel frameworks;

Cons:

- High result dependency on the approach of cluster center initializations;
- Requires additional verification of the robustness of the final clusters;

2.1.2 Hierarchical methods

All clusters form a hierarchical structure that can be organized in top-down or down-top orders. The hierarchy is created using the following components:

- **The similarity matrix** consists of the similarities between each pair of units. The metric that is used to construct this matrix impacts the configuration of the final clusters;
- **Linkage criterion** b– a function that determines the distance between clusters and allows to link them in the hierarchical structure;

Pros:

- The possibility of obtaining good results by selecting an appropriate similarity function;

Cons:

- Cubic $O(n^3)$ time and quadratic $O(n^2)$ space complexity
- Counterintuitive nature of distance between two clusters in one hierarchical node

2.1.3 Model based methods

In the context of this cohort of algorithms, the "model" denotes the user-defined statistical distribution that each observation should match.

Pros:

- Accessible parameters tuning in the underlying user-defined function for each particular problem;

Cons:

- The slower model performance itself due to the high complexity;

2.1.4 Neural network (NN) based methods

The majority of studies using NN for clustering concentrates on two approaches:

- **Self Organizing Maps (SOM)** - a NN that nonlinearly projects data onto a lower-dimensional space of features where cluster analysis can be performed;

- **Adaptive Resonance Theory (ART)** - the combination of supervised and unsupervised learning methods to solve the pattern recognition problem in a way that is similar to how the brain processes information;

In both approaches, qualitative features are firstly transformed into binary ones that later are treated as numerical values

Pros:

- APT predictions are resistant to the changes in a dataset;

Cons:

- SOM may result in low-quality topological mappings;
- Due to the usage of differential equations, ART models may have high computational complexity;

2.1.5 Other

Such methods as spectral, subspace, or density-based clustering cannot be allocated to any of the cohorts above but can still be used for clustering mixed data. They are only listed here and will not be described in detail further. However, with comprehensive coverage of different clustering algorithms, Ahmad and Khan's paper lacks the example with results of the application of those algorithms for solving real problems.

Chapter 3

Data

3.1 Overview of the online marketplace

As a source of customers' data in this research, we will be using the online marketplace JustAnswer which provides a communication service with experts in different fields. The company has been operating since 2003 and is represented in 5 countries - the United States of America, the United Kingdom, Spain, Germany, and Japan.

The business model of JustAnswer is the following. A customer visits the platform for the first time with a request that the experts can fulfill. After a brief conversation and a description of the question itself, the customer is proposed to buy a monthly subscription to proceed to direct communication with an expert. If a successful conversion happens, the customer is waiting for a connection with a professional with appropriate expertise. With a subscription, the customer can ask as many questions as is needed during the next month. If the customer does not cancel the subscription within the month, he will be charged monthly for access to the platform.

JustAnswer is facing a common challenge for online-service providers - the problem of aggregating their customers into meaningful groups to understand the needs of each such group better. During the company's long history, a well-designed data infrastructure was developed to track many data from processes happening in the marketplace. The sample from this data will be used in this paper.

3.2 Dataset overview

Consistent with the topics of this work, we will have the customers' heterogeneous data with both quantitative and qualitative features. Dataset is anonymized, and all numeric variables are changed to hide the sensitive business values and reflect the overall data structure.

The original dataset consists of 97,965 observations, each representing one unique customer. The period of data is April 2022, and all customers in data got to the site for the first time through paid advertising. The example of content is provided in the table below:

ID	Device	Category	Browser	Operating System	35D LTV	CPA
1	Mobile	Law	Chrome	iOS	58.54	2.16
2	Mobile	Computer	Safari	iOS	2.3	96.68
3	Desktop	Law	Chrome	Windows	112.93	2.94

TABLE 3.1: Sample from the original data set

- **Id** - unique customer identifier
- **Device** - a type of device that the customer used during a conversion
- **Category** - a field in which the customer asked his first question
- **Browser** - an internet browser that the customer used during a conversion
- **Operating System** - a system of the customer's device
- **35d LTV** - **L**ife**T**ime **V**alue of a customer during the first 35 days of platform usage
- **CPA** - **C**ost **P**er **A**cquisition - the marketing costs associated with the acquisition of a customer

3.2.1 Categorical features

Since numerical metrics cannot measure the qualitative variables, we will show the spread of customers among different variables of such a type instead. One of the main characteristics of each customer is the category of the question that led this person to the platform. As shown in Figure 3.1, the most significant share of data is allocated in the Computer, and among other categories, the share size is relatively evenly.

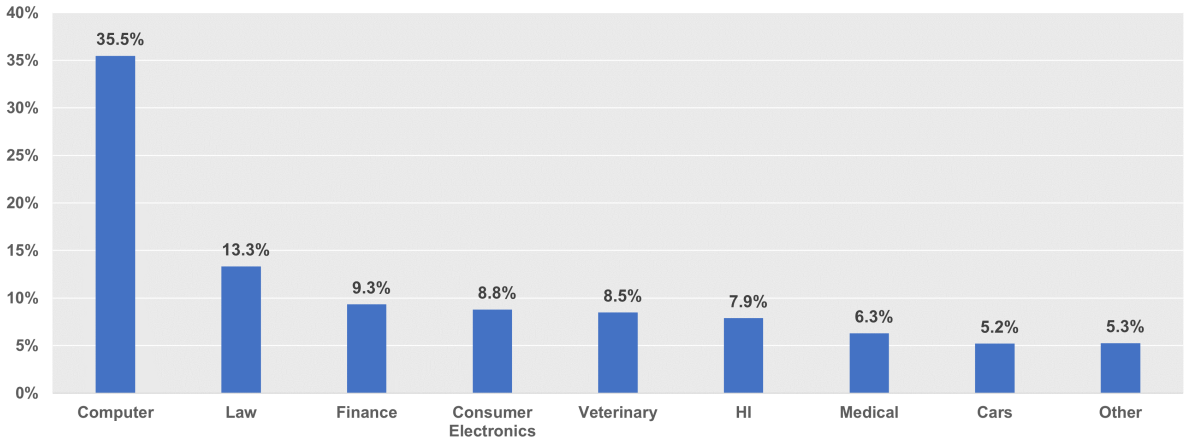


FIGURE 3.1: Distribution of Customers by Category

Another critical feature of customers is the device since, depending on it, customers receive different user experiences. Based on Figure 3.2, we can see that the most popular device is the Mobile, and the second in popularity is the Desktop:

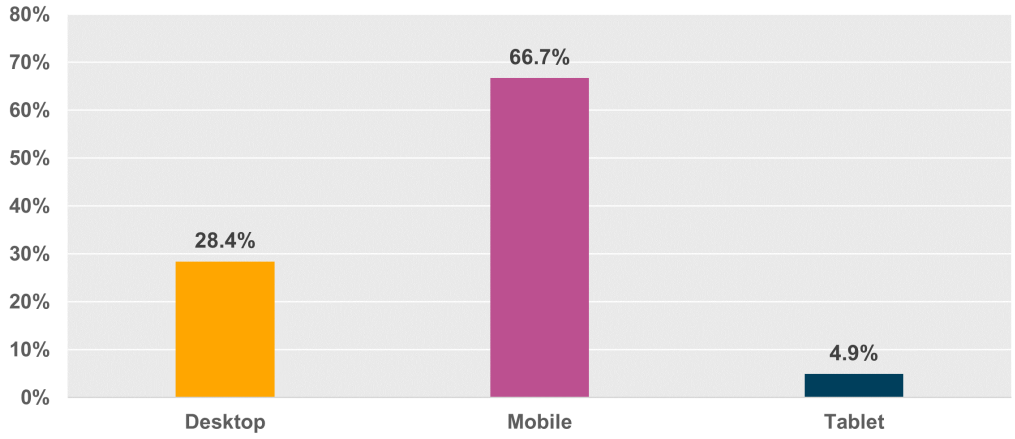


FIGURE 3.2: Distribution of Customers by Device

To better understand the data set structure, we show the combination of these two factors in Figure 3.3. It tells us that the general spread of data by a device is typical for most categories. Only in HI (Home Improvement) the difference between shares of Desktop and Mobile is smaller than in other categories.

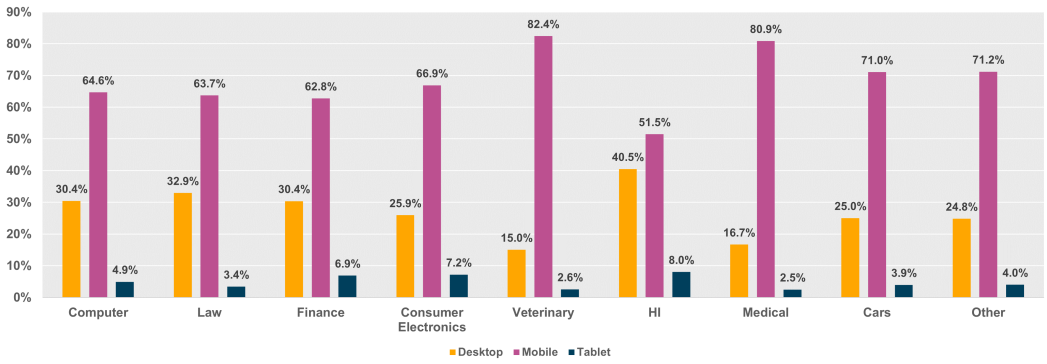


FIGURE 3.3: Distribution of Customers by Category and Device

One more feature that should be considered in describing the structure of customers is the type of operating system they are using. This characteristic can implicitly represent a lot of customer actions and decisions. From Figure 3.4, it is clear that mainly customers are using the iOS/Android system.

However, such a conclusion can be wrong due to the underlying mix of devices. We added figures representing the same distribution but separately for each device to avoid such a mistake. Now, it is clear that the previous conclusion about iOS/Android was proper for Mobile users. Nevertheless, among Desktop users, such systems as Windows and Macintosh are prevailing.

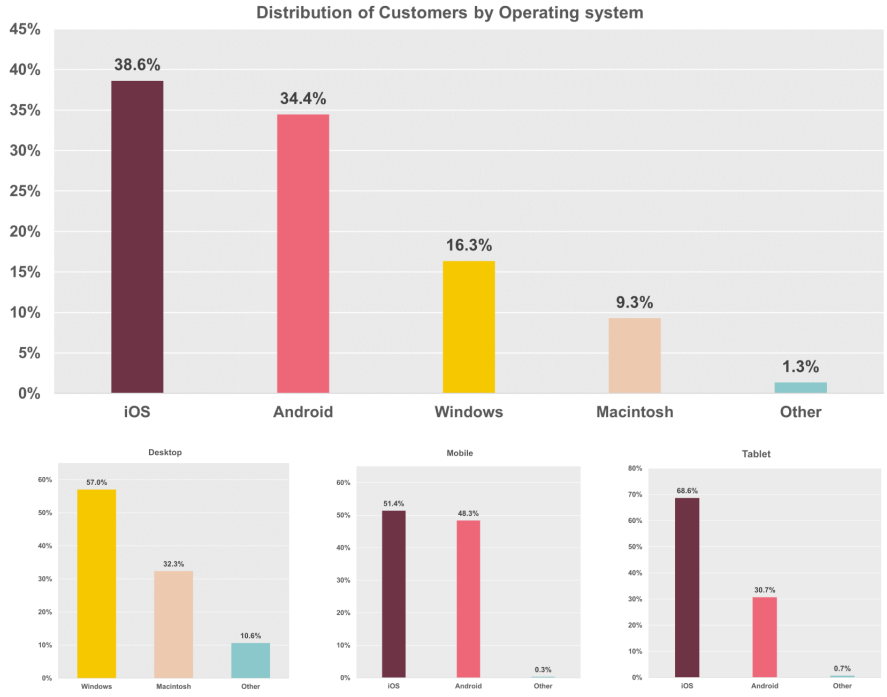


FIGURE 3.4: Distribution of Customers by Operating System and Device

3.2.2 Continuous features

In the dataset, there are two quantitative variables - CPA and 35D LTV - that respectively show the cost and revenues related to customers. Both of them are given in US dollars.

CPA always has positive value and is dynamically defined by the paid traffic provider. There are many external factors influencing the value of CPA, such as customer requests, but they will not be considered in the scope of this work.

LTV represents a company's money amount from a customer and depends on his behavior. The value of this metric can be either positive or negative. The negative LTV appears in the case when a customer requests a refund of his money, and the company returns them. The general overview of these two metrics is provided in table 3.2

Metric	35D LTV	CPA
Min	-50.85	1.94
1st. Qu.	20.22	3.08
Median	30.73	5.55
Mean	33.41	33.52
3rd Qu.	48.03	21.09
Max	421.94	1355.3

TABLE 3.2: Continuous features summary

To provide the reader with a better understanding of the underlying trends in data, we show the distribution of continuous variables combined with the categorical variables discussed above. For better perception of the visualization, the value of LTV was limited to 200$, and the value of CPA was limited to 40$.

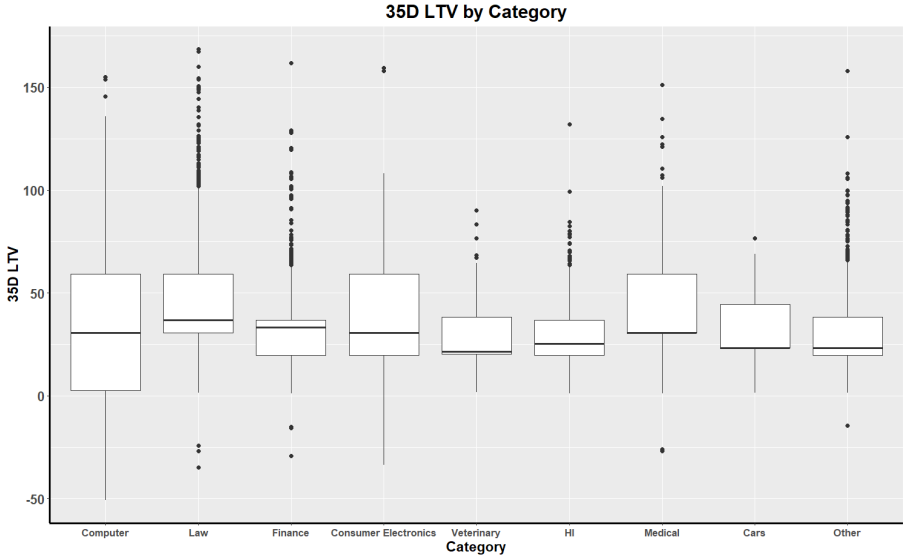


FIGURE 3.5: 35D LTV by Category

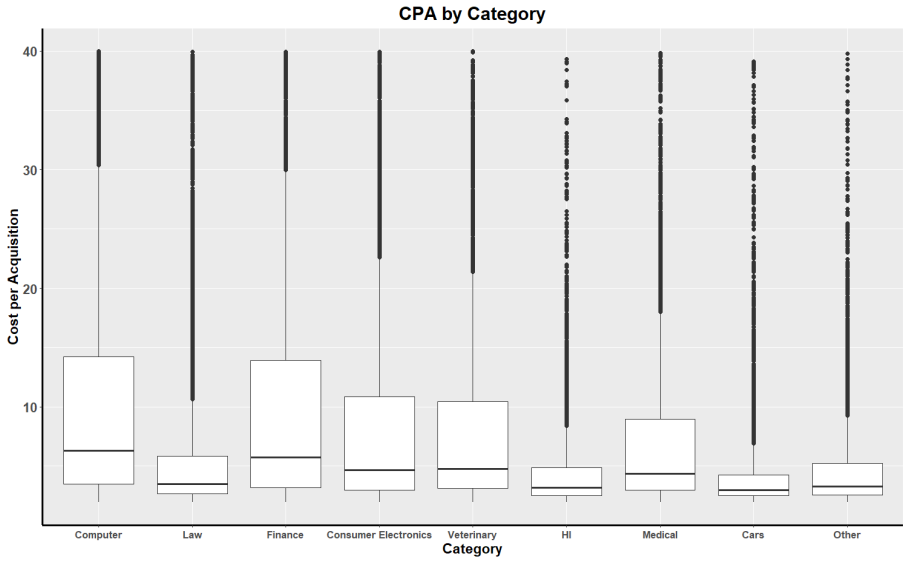


FIGURE 3.6: CPA by Category

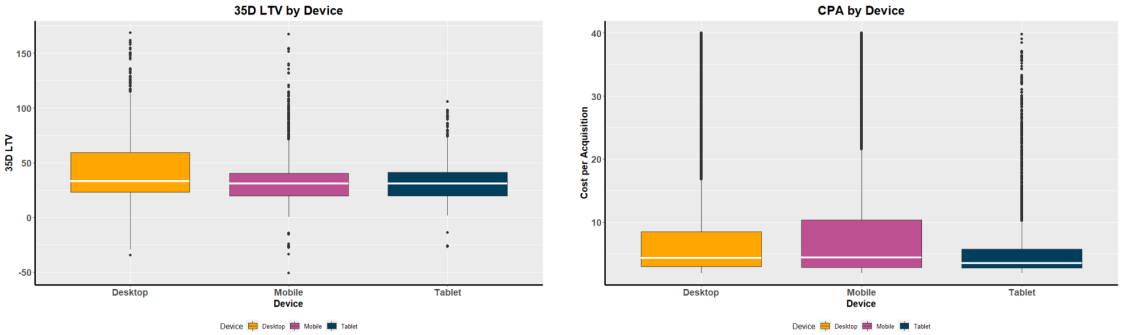


FIGURE 3.7: Continuous features by Device

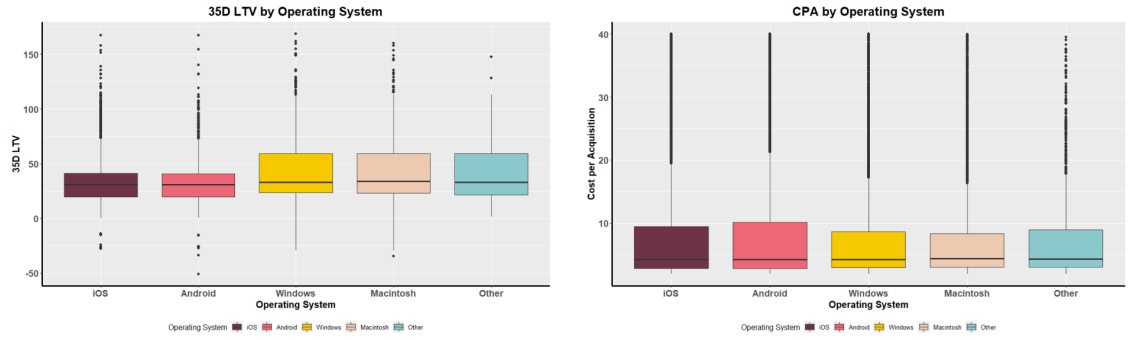


FIGURE 3.8: Continuous features by Operating System

Chapter 4

Methodology

In this chapter, we will cover the clustering methods used in the research and techniques for measuring the quality of the results and their visualization. Based on the papers covered in the previous chapter, we decided to select clustering methods from three groups - partitional, model-based, and neural network-based approaches. A detailed description of each technique will be provided in this chapter.

4.1 Clustering methods

4.1.1 Model-based clustering methods

KAMILA

One of the well-known models used in cluster analysis is the k-means. However, it cannot be applied to the heterogeneous data since it is oriented only on the continuous variables. To handle different types of data, a modification of this model called **KAMILA** - **KA**y-**M**eans for **MI**xed **LA**rge data - was introduced by Foss&Markatou [5]. The main difference between the original implementation and the modified version is incorporating the term for handling categorical variables of each observation in the core function.

The **kamila** algorithm follows the next logic: having $\boldsymbol{N}$ independent and identically distributed data points, each of which is the vector of dimensions $(\mathbf{P} + \mathbf{Q})$, where $\mathbf{P}$ is the dimensionality of vector $\mathbf{V}$ of continuous random variables and $\mathbf{Q}$ is the dimensionality of vector $\mathbf{W}$ of categorical random variables. Each observation $i \in \boldsymbol{N}$ must belong to the one of the components $\mathbf{g}$ from the set of cardinality $\mathbf{G}$. At the start of algorithm execution for each $\mathbf{g}$, the following parameters are estimated:

- μ_g - centroid of population $\mathbf{g}$. The initial value is randomly drawn from the existing dataset;
- θ_{gq} - a vector with parameters of the multinomial distribution of the q-th ($q = 1, 2, ..., \mathbf{Q}$) categorical variable.

After that, on each iteration for each observation $\mathbf{i}$ and cluster $\mathbf{g}$, distances for quantitative features and probabilities for qualitative features are calculated.

The Euclidean distance **d** from the centroid g to point **i** is calculated by the following formula 4.1:

$$d_{ig} = \sqrt{\sum_{p=1}^{P} \left[(v_{ip} - \hat{\mu}_{gp}) \right]^2} \tag{4.1}$$

where $v_{ip} \in \mathbf{V}$ - one of the continuous variables of each unit.

Instead of distance, for qualitative features, the authors are calculating the probability **c** of observing the set of categorical variables of observation **i** within population **g** by the formula 4.2 given below

$$c_{ig} = \prod_{q=1}^{Q} m(w_{iq}; \hat{\theta}_{gq}), \tag{4.2}$$

where $m(\cdot;\cdot)$ - the multinomial probability mass function

w_{iq} - one of the categorical variables of each unit.

At the end of each iteration, each point **i** is assigned to the cluster **g** in the way that maximizes the objective function **H**:

$$H_i(g) = log\left[\hat{f}_{\mathbf{V}}(d_{ig})\right] + log\left[c_{ig}\right] \tag{4.3}$$

where $\hat{f}_{\mathbf{V}}$ - kernel density estimate of the minimum distance.

The parameters μ and σ are recalculated for the subsequent iterations at the end of the iteration using updated cluster assignments.

The algorithm is stopped at the moment when clusters remain the same for the pair of consecutive iterations. The cluster assignment with the highest objective function value among all iterations is chosen as the final one.

Latent Class Model

Another model-based clustering method is the Latent Class Model (**LCM**) by Clogg [4]. The main idea of this model is the presence of latent variables in the dataset. They can be hidden in the combination of two or more variables. Knowing that the joint distribution can express one latent variable, all variables that belong to the one latent variable are grouped in the one latent class.

LCM concentrates on finding such classes in the mixed data consisting of both continuous and categorical features and using them to cluster the data points.

The LCM method works in the following way: We define the set of latent variables **X** consisting of **T** categories. Having the dataset with **N** observations and assuming that all variables in **N** are independent, we can calculate the joint distribution of the **N** and **X**:

$$\pi_{\mathbf{N},\mathbf{X}}(i,t) = \pi_{\mathbf{X}}(t)\pi_{\mathbf{N}|\mathbf{X}_{(t)}}(i) \tag{4.4}$$

where i - observation from $\mathbf{N}(n = 1, 2, ..., \mathbf{N})$
t - category from $\mathbf{T}$ $(t = 1, 2, ..., \mathbf{T})$
$\pi_{\mathbf{X}}(t)$ - latent class $\mathbf{t}$ proportion
$\pi_{\mathbf{N}|\mathbf{X}_{(t)}}(i)$ - conditional density function for each feature in $\mathbf{N}$
And the final form of the LCM can be denoted as done in 4.5

$$\pi_{\mathbf{N}}(i) = \sum_{t=1}^{\mathbf{T}} \pi_{\mathbf{N}|\mathbf{X}}(i,t) \tag{4.5}$$

In this work, we will be using the modification of LCM call VarSelLCM, developed and released as an **R** package by Marbac et al [9]. The authors implemented the existing model using an adaptation of the EM algorithm [6] to perform the feature selection using both Bayesian information criterion and maximum likelihood inference at the same time.

In the first steps of the algorithm, we initialize the following parameters:

- $\boldsymbol{G}$ - a mixture of components (clusters);
- $\boldsymbol{w}$ - binary vector of cardinality $\boldsymbol{d}$ showing if feature w_j is relevant or not, where $\boldsymbol{d}$ is the number of features in each observation;
- $\boldsymbol{S}$ - a matrix that defines the partition of data points by clusters. Each of the N columns represents observation i in the data set, and each of the G rows represents if observation i belongs to cluster $g \in G$.

Parameters G and w together form the estimated model $\hat{m}$, and the matrix S is used for the maximum likelihood estimator $\hat{\theta}$.

On each iteration of the EM algorithm, in step E, we calculate conditional probabilities $t_{ik}(m, \theta)$ for each cluster g in S. In step M, we maximize the expectation complete-data log-likelihood using BIC penalizing.

The common challenge during clusterization is when the number of features is higher than the number of observations in the data. It is better to use the Maximum Integrated Complete-data Likelihood (**MICL**) in such a situation. This criterion is based on the closed form of the integrated complete-data likelihood [9]. Also, it is oriented on clustering as the final result of the technique, making it more relevant to the purposes of this research.

One of the potential drawbacks is the convergence to the local optima and possibly missing the best result. It is solved in the VarSelLCM by automatically random initialization for different algorithm runs.

Latent Class Analysis

The Latent Class Analysis (**LCA**) method, like the previous technique LCM, comes from the Latent Structure Analysis[Lazarsfeld PF (1950)]. LCA realizes the Latent Cluster Regression methodology developed by Bandeen-Roche et al. [2].

The main difference between LCA and LCM is that LCA treats each variable in the dataset as a categorical one (from here and further in the text, they will be called "manifest" variables [poLCA R package paper]). Such an approach for processing input data challenges working with mixed data since such datasets also contain continuous variables. Gregoire Preud'homme et al.[Nature paper] proposes discretizing quantitative features based on the percentiles to have percentiles values as levels of each such feature.

Another assumption in LCA is that all manifest variables are sampled from the combination of the multinomial distributions defined by the assignment of observations of each feature to the clusters (from here and further in the text, they will be called "Latent Classes").

As the programming realization of this method, we will be using the poLCA **R** package by Linzer et al. Implementation of LCA by the authors uses EM and Newton-Raphson algorithms [Linzer].

The following steps can describe the general LCA algorithm. We denote the expected size of the Latent Class $g \in G$ as a proportion τ_g [Nature paper], where $\tau_g \in (0,1)$ and a $\sum_{g=1}^{G} \tau_g = 1$). Initial values of τ are sampled from the uniform distribution. Each g is defined by the probability distribution function with the set of parameters α_g. Together sets of τ and α form the matrix θ that describes each Latent Class.

Having that, the density function of the observation $i \in N$ is calculated by the formula 4.6 below:

$$f(i|\theta) = \sum_{g=1}^{G} \tau_g \, h(i|\alpha_g) \tag{4.6}$$

where $h(i, \alpha_g)$ - distribution function of g

The objective function of the LCA is the maximization of the log-likelihood function by the EM algorithm with a Newton-Raphson step [Bandeen-Roche]:

$$ln\ L = \sum_{i=1}^{N} ln\Big(\sum_{g=1}^{G} \tau_g \, h(i|\alpha_g) \Big) \tag{4.7}$$

The algorithm is stopped when the difference between the results of two successive iterations is less than the defined tolerance value (by default, in the poLCA, it is equal to the $1 * 10^{-10}$ [poLCA package]). Another condition of algorithm stop is the reaching the maximum number of iterations. The assignment to Latent Classes with the highest value of 4.7 is returned as a final clustering.

4.1.2 Partitional methods

Partitioning Around Medoids

The first one of the partitional methods is the Partitioning Around Medoids method [Kaufman]. This algorithm searches for G of the most representative objects among data points called medoids. Those objects are used to create G of the clusters. The name of the cluster-defining object differs from the conventional name "centroid" because medoids are constructed not based on distances between units but on similarities between them. This feature of PAM allows us to use it with mixed data.

Another essential feature of this technique is the option to provide the input data not as a data set with the exact value of each variable but as a matrix of similarities between all observations in data. The approach used for similarity calculation will be discussed later in section XX.

The underlying logic of the PAM method is minimizing total dissimilarity between medoids $(m_1, m_2, ..., m_G)$ and points in the respective clusters they forms. This objective function can be expressed in the following way:

$$TD = \sum_{g=1}^{G} \sum_{i \in C_g} d(i, m_g) \tag{4.8}$$

where C_g - set of all points assigned to cluster g
$d(i, m_g)$ - similarity between observation i and medoid m_g.

The PAM clustering algorithm consists of two parts - BUILD and SWAP. In the BUILD part, the G data points with the lowest value of the sum of dissimilarities to the rest of the points are selected as initial medoids.

In the SWAP part, medoids from the previous step are swapped with all non-medoids to find the cluster configurations that minimize the objective function. The set of medoids and points assignments to them that results in the minimum value of 4.8 is returned as the final clusters.

KPrototypes

This method was developed by Huang [7] as a hybrid of two algorithms - K-means (process only quantitative variables) and K-modes (process only quantitative variables). The object that is the center of each cluster is called Prototype. These object are constructed as a combination of two types of features. Also, there is an option of regulating the influence of different type of variables using the parameter γ.

4.1.3 Neural network based methods

Self Organizing Maps

The one clustering method that is neural network-based and will be discussed in this work is the Self Organizing Maps that were introduced by Kohonen [10]. The logic behind this technique is the following:

1. For each data point i, we are calculating the "Best Matching Unit" (BMU) - another point in a dataset that is the most similar to the i;
2. Define all points in the neighborhood with BMU. The number of those points changes on different iterations;
3. On each iteration, for all points in the neighborhood, assign a weight in such a way that only the closest points remain;

The critical entity in the SOM is the map itself. For our purposes, we will define maps as one-column or one-row maps containing the number of cells that is equal to the number of clusters.

4.2 Other techniques used in the research

4.2.1 Gower coefficient of similarity

One of the fundamental techniques in this cohort of methods is the general coefficient of similarity developed by John Gower (1971). Instead of measuring distance only for quantitative variables, Gower proposes calculating the similarities for each unit's binary, qualitative, and quantitative features separately and combining the obtained values in one coefficient representing the general similarity between two units ranging between 0 and 1. Denoting Gower's similarity for two observations x and y as $S(x, y)$, the formula will be the following:

$$S(x,y) = \frac{\sum_{k=1}^{K}(s_{xyk} * w_{xyk})}{\sum_{k=1}^{K} w_{xyk}} \tag{4.9}$$

where K - the total number of features in each unit

s_{xyk} - similarity of x and y using feature k

w_{xyn} - binary weight that indicates the possibility of comparing x and y using feature k. When $\sum_{k=1}^{K} w_{xyk} = 0$, then two units are not comparable, and $S(x, y)$ is undefined.

The challenge that one has to consider is the time and space complexity. Since the Gower similarity is computed for each pair of observations, the time complexity is the $O(n^2)$. The result of this technique is the matrix that contains the value of the coefficient for each pair from the previous step, and thus the space complexity is also $O(n^2)$. Calculations of Gower similarity were performed using cluster **R** package [link to cran article].

4.2.2 Silhouette score

The problem the one who tries to cluster mixed data must solve is to measure the quality of the result of the clustering method. It is complicated because there is no way to get the truth partitioning of customers by clusters in the selected business domain. One of the solutions that can be used here is the Silhouette score (Rousseeuw

1987) which shows how well data points are allocated between clusters. The score value for each n from N observations in the data and cluster $g \in G$ to which n belongs is calculated by the following steps:

1. Calculate the average similarity between n and all other observations in the cluster g. Denote the result as $a(n)$.

2. For all other clusters $g' \in G$ except g, calculate the average distance between n and all points in the cluster g'. Take the smallest one among those distances and denote it as $b(n)$.

3. The silhouette score **s(n)** for n will be the following:

$$s(n) = \frac{b(n) - a(n)}{max[a(n), b(n)]} \tag{4.10}$$

   s(n) is always defined in the interval [-1; 1], and the higher value of s(n), the better the cluster assignments.

4.2.3 Calinski-Harabasz Index

Another tool of the validation of cluster results is the Calinski-Harabasz Index [3] also called as Variance ratio criterion. The logic of it is similar to the Silhouette Score: this index measures the similarity between the cluster to which point is assigned and compares this value for the analogue for all other clusters.

4.2.4 tSNE

The third way of validating the clustering results is to visualize them. Nevertheless, when dealing with the high dimension mixed data, it becomes difficult to present the cluster assignments. The algorithm solving this problem is the t-distributed Stochastic Neighborhood Embedding (tSNE) [] that allows reducing dimensionality to 3D or 2D. The tSNE technique consists of 3 steps:

1. In the original paper by Laurens&Hinton[], in the first step, the authors suggest calculating the similarity matrix using the Euclidean distance. We will omit this part in our work since we will be using the distance matrix obtained by using the Gower coefficient described above. To follow the notation of the authors, the value of similarity between two observations n_i and n_j will be denoted as a p_{ij}.

2. In the second step, we create the counterparts y_i and y_j in the low-dimensional space for n_i and n_j, respectively. Having these points, we can calculate the joint probabilities for them as follows:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}} \tag{4.11}$$

Also, this value can be interpreted as a similarity between n_i and n_j in the low-dimensional space [Laurens].

The main advantage of the tSNE is using a t-distribution instead of Gaussian in this calculation. The "heavy tails" property of t-distribution solves the so-called "crowding" problem. This problem appeared in the basic version of tSNE - SNE [SNE authors link] - when the medium similarity between two observations in the original dataset results in a "crowd" of observations with approximately the exact similarity. The "heavy tails" of t-distribution place such points far away in the two-dimensional space, making comprehension of the visualization much more straightforward.

3. With values from two previous steps, in the third step, we minimize the Kullback-Leiber divergence between joint probability distributions P (high dimensionality) and Q (low dimensionality):

$$C = KL(P\|Q) = \sum_i \sum_j p_{ij}\, log \frac{p_{ij}}{q_ij} \tag{4.12}$$

Minimization is performed by the changing values in the lower dimension data to make Q similar to P. The logic here is the following: having the similarity of each pair of points in the Q as identical as possible to the same values in P will bring us the low dimension representation of data from the original dataset. The implementation of tSNE in **R** was taken from package Rtsne [].

Chapter 5

Results

This chapter will describe the application of all methods and techniques from the Methodology chapter to solve the problem of clustering customers represented by mixed data. Also, we will cover the challenges that we faced during this process and their solutions. All computations were performed on a machine with i7-10850 CPU @ 2.7GHz and 32GB of RAM.

5.1 Description of the general algorithm

Instead of the distance between two data points, we measured their similarity using the Gower coefficient. From the results of these calculations, we constructed a diagonal similarity matrix where each cell shows the similarity between each pair of observations in the data set.

Here is an example of such calculations. For the data sample in table 3.1 , the similarity matrix would be as shown in table 5.1

ID	1	2	3
1	0.0		
2	0.58	0.0	
3	0.42	0.99	0.00

TABLE 5.1: Similarity matrix for the sample of data in Table 3.1

Having this matrix, we visualized all data points in the 2D space by the tSNE technique. The 2D reflection of similarities without clustering is provided in figure 5.1:

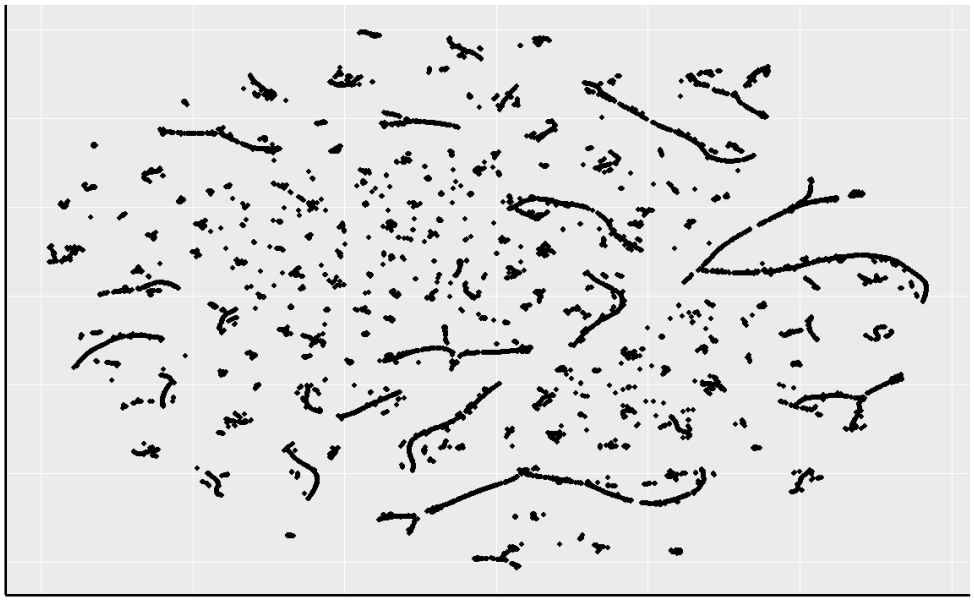


FIGURE 5.1: Customers visualization in 2D space by tSNE without clustering

5.2 Definition of the size of the data set

As we mentioned in chapter Data, the raw data contains 97,965 rows. If we want to calculate the similarity matrix for all observations, we will need the number of rows given below:

$$\binom{n}{2} = \frac{n^2 - n}{2} = \frac{97965^2 - 97965}{2} = 4\,798\,521\,630$$

Storing and working with such a big object can be very costly in terms of required memory and computational time. Also, some R packages used for the algorithm implementation cannot process objects of such size.

For example, the function ***daisy()*** from the package **cluster** for computing the Gower coefficient limits the dimensionality of the input data frame by the value $2^{16} = 65536$. Moreover, the function Rtsne() from the package Rtsne is raising the error. With the empirical tries, we figured out that the biggest possible number of observations in the input data is around 45000.

To meet this requirement, we decided to reduce the number of observations. So, we randomly sampled original data, created 3 data frames with 45000, 15000, and 5000 rows, and executed the algorithm for each. During that process, we faced the fact that the algorithm run time for the data frame of size 45000 is enormously big - more than 72 hours. Due to this obstacle, we proceed only with datasets of smaller sizes. The detailed outcomes will be discussed only for the data frame of 15000, and the general results for both data frames will be compared at the end of the chapter.

5.3 Determining the best number of clusters

The vital problem in each clusterization problem is finding the best number of clusters. We defined the optimal number of clusters for each clustering method in the following steps. It was done by the following approach:

Algorithm 1 The best number of clusters selection

```
for each m in methods do:
    for c = 1, 2, ..., 8 do
        p ← Partition data by method m with number of clusters c
        Calculate the Silhouette score for partition p
        Calculate the Calinski-Harabasz Index for partition p
    end for
    Select the best value of clusters c for method m
end for
```

The algorithm described was executed for all selected clustering methods, and its results are given later.

In our work, we use the combination of two parameters - Silhouette Score and Calinski-Harabasz Index. Using them in a pair is beneficial in situations when one of the indexes has equal value for a different quantity of clusters. In this case, the value of another index can be used to make a final decision.

Such an incident we experienced in our work also. In figure 5.2, there are line charts with values of both indexes for methods KPrototypes and LCM.

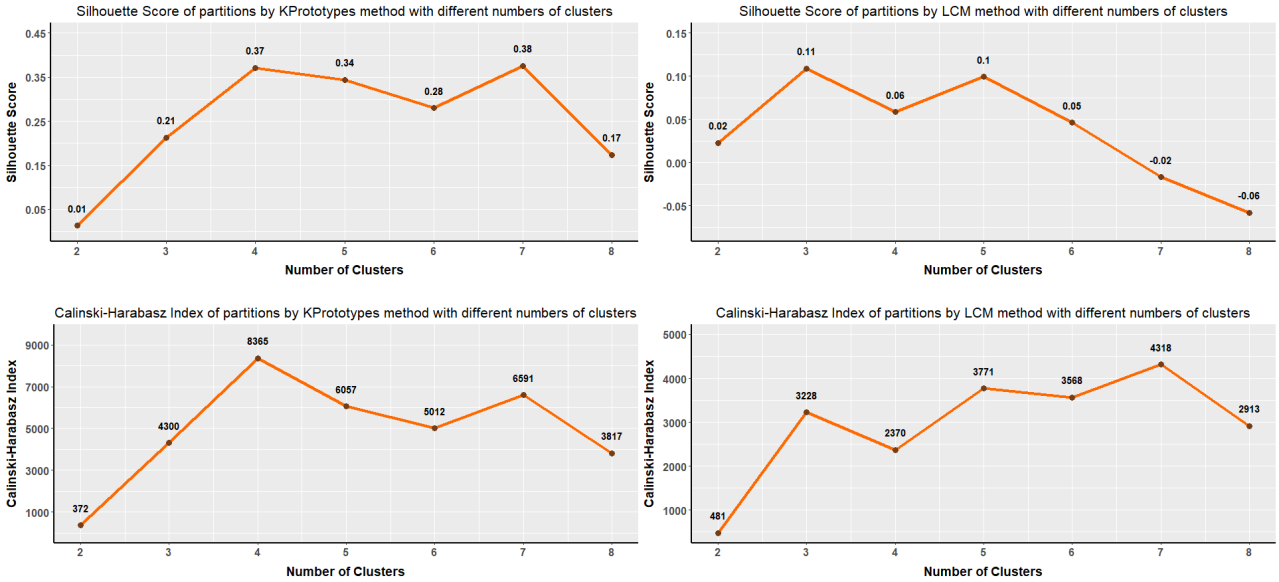


FIGURE 5.2: Clustering quality indexes for KPrototypes and LCM methods

We can see that for the KPrototypes method, Silhouette Score for 4 and 7 clusters are almost the same - 0.37 and 0.38. Nevertheless, Calinski-Harabasz Index tells us

that its value for 4 clusters is much higher than for 7. So, we should choose the option with fewer clusters.

Moreover, a similar case is for the LCM method. Calinski-Harabasz Index is the highest for the 5, 6, and 7 clusters. After comparing values of the Silhouette Score for the same numbers of clusters, we can conclude that partitioning into five groups is the optimal solution for the LCM method.

The charts of this type for all methods are available in Appendix A.

5.4 Comparison of the results

When the best set of parameters of each model is defined, we can proceed with selecting the final methods for clustering. The final number of clusters and Silhouette score values for each method and size of the data set are presented in Table 5.2. We do not provide Calinski-Harabasz Index values because they heavily depend on the number of observations and will not be relevant in this context.

Method	The best number of clusters		Silhouette Score	
	5k	15k	5k	15k
PAM	4	4	0.52	0.52
LCM	4	5	0.2	0.1
LCA	3	3	0.5	0.5
KPrototypes	5	4	0.4	0.37
KAMILA	7	6	0.18	0.26
SOM	4	4	0.44	0.49

TABLE 5.2: Clustering quality indexes for all methods and datasets sizes

As we can see, the PAM and LCA have the best Silhouette Scores among all methods. So, we will use them to form the segments of customers.

Another important outcome from this table is that model configuration and Silhouette Score stay stable for different dataset sizes. Based on that, we state that there is no need to use big data sets to obtain reliable results.

5.5 Explanation of the clusters

In this section, we explain the clusters that were defined by the two methods selected before.

5.5.1 PAM method

PAM method used three variables - Device, Browser, and Operating system - as the pivots for constructing the four customer segments. The summary of each of them is provided in table XX.

We interpret obtained clusters in the following way:

- Cluster **iOS users**:

  In this group, among operating systems, iOS dominates. We can see that share of desktop users is the most minor, and the majority are using devices such as Mobile or Tablet on which iOS can be installed.

- Cluster **Desktop Chrome users**:

  We can see that almost all customers are desktop users in this segment. Nevertheless, they do not use solely one particular system. Because of this fact, we decided to interpret this cluster like the one with Chrome browser users.

- Cluster **Android users**:

  The logic of this cohort is pretty similar to the first one. We have in this group one dominating device - Mobile - and one prevailing system among operating systems - Android.

- Cluster **Macintosh users**:

  The main characteristic of this cluster is that almost all customers have Macintosh as their operating system.

In addition to the information above, in Figure 5.3, we show the distribution of 35D LTV by each cluster. Since the structure of clusters heavily depends on a customer's device type, the distribution in Figure 5.3 reflects a similar solely for devices provided previously in Figure XX. Along with LTV distribution, in Figure 5.4, we visualize the cluster assignment in the 2D space obtained from applying the tSNE technique.

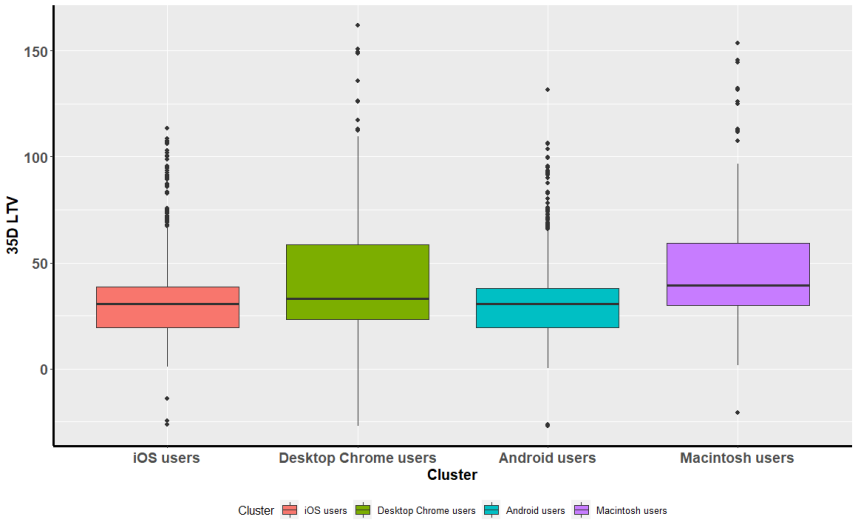


FIGURE 5.3: 35D LTV by Clusters assigned by PAM method

FIGURE 5.4: Customers visualization in 2D space by tSNE with PAM clustering

5.5.2 LCA method

In LCA's partition, there are three clusters based on two variables - Device and Operating system. A description of each of them is given below.

- Cluster **iOS/Android users**:

  We merge the description of these two clusters since the main difference between them is the type of system on a device. In these two groups located almost all Mobile/Tablet users.

- Cluster **Desktop users**:

  As all non-Desktop customers were allocated to the two previous groups, in this one, we have mainly only those people that are using Desktop. All possible variants of values of features "Operating system" and "Browser" are hidden in this cluster.

The summary of all variables for each cluster is provided in table XX, and the graphs with LTV distribution and 2D visualization are provided in Figure XX. The same graphs for all other methods are available in Appendix B.

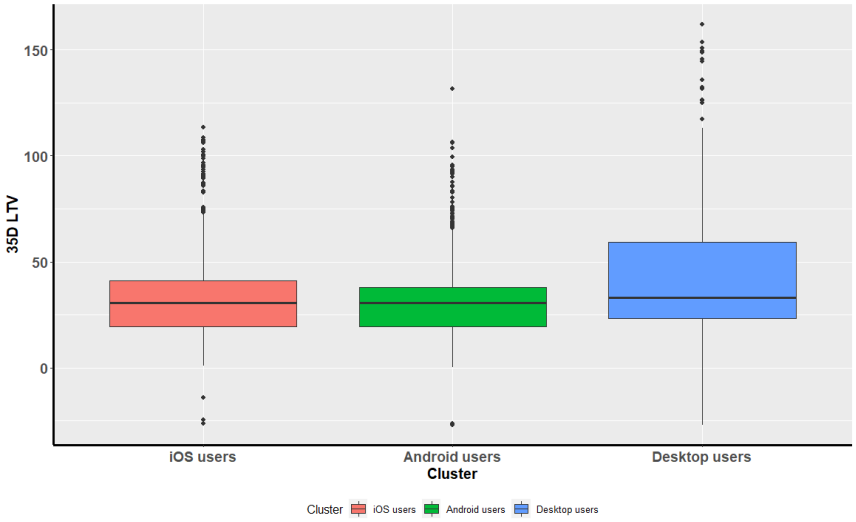


FIGURE 5.5: 35D LTV by Clusters assigned by LCA method

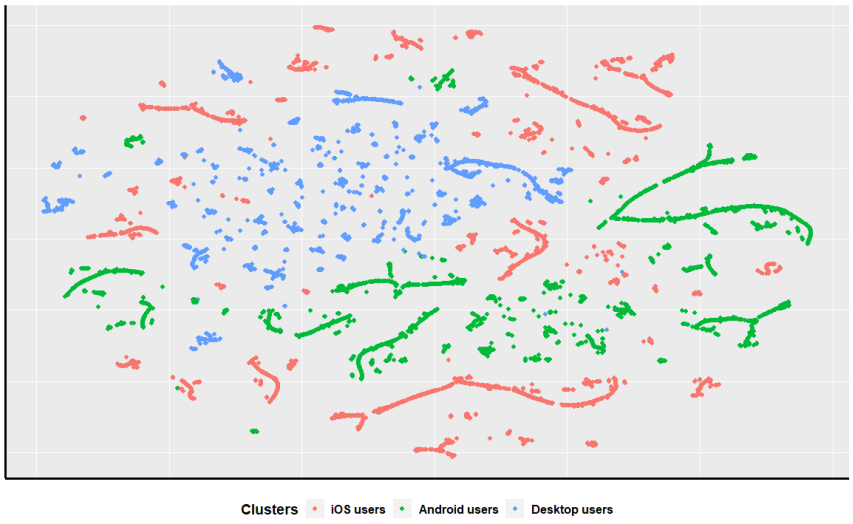


FIGURE 5.6: Customers visualization in 2D space by tSNE with LCA clustering

Chapter 6

Conclusions

The main goal of this bachelor thesis is to define the best clustering method that allows for obtaining meaningful groups of customers from heterogeneous datasets. In the scope of this problem, we defined two tasks - to carry out a comparison of methods for clustering mixed data and to apply the best of them for online-marketplace customers' data.

As a summary of the fulfillment of the first goal, we can state that many techniques are developed for clustering mixed data. Moreover, they form 5 groups of algorithms - partitional, hierarchical, model-based, Neural network-based, and others. From the variety of existing solutions, we selected six methods:

- KAMILA - KAy-Means for MIxed LArge data;
- LCM - Latent Class Model;
- LCA - Latent Class Analysis;
- PAM - Partitioning Around Medoids;
- KPrototypes
- SOM - Self-Organized Map

We trained each of these models with a different number of clusters. The next step was to evaluate the Silhouette Score and Calinski-Harabasz Index results to choose the best. Furthermore, as the outcome, we decided to use the PAM method with 4 clusters and the LCA method with 3 clusters.

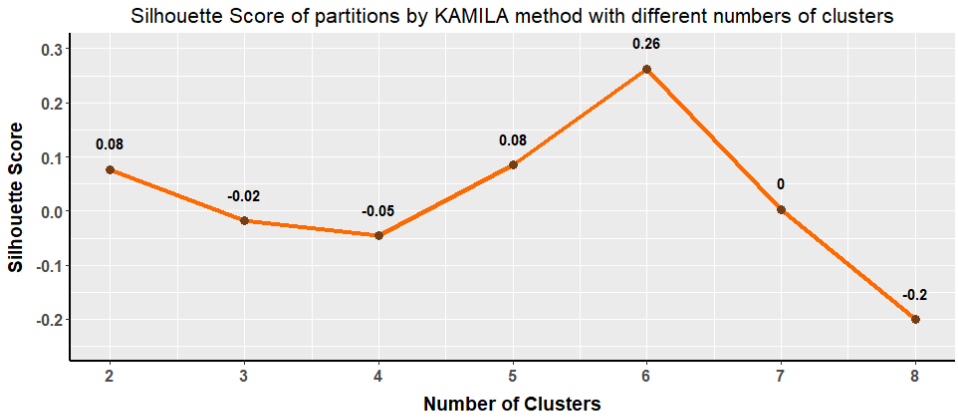
To reach the second goal, we used the customers' data from the online-marketplace JustAnswer. Since there are no true labels of customer clusters in this field of business, this case fits the unsupervised learning problem that we want to solve.

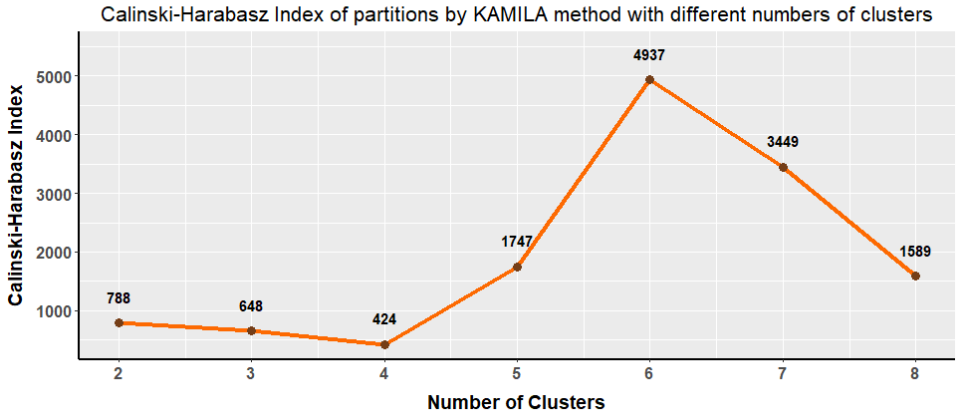
In conclusion, we state that conducted research provides practical tools for solving the selected business problem. Another essential characteristic is its versatility since it can be applied to any dataset, regardless of which business area it comes from. Moreover, in a combination of the strong domain knowledge, the outcomes in the form of customer segments can provide valuable and actionable insights for the business stakeholders.

Appendix A

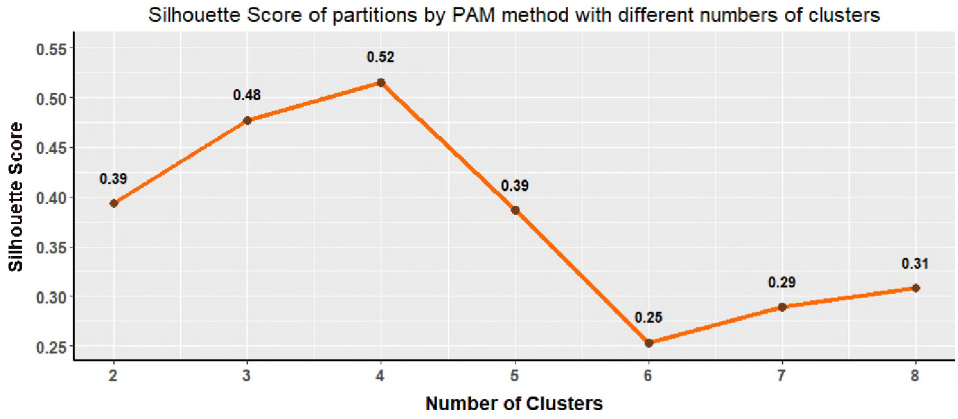
Clustering methods quality metrics

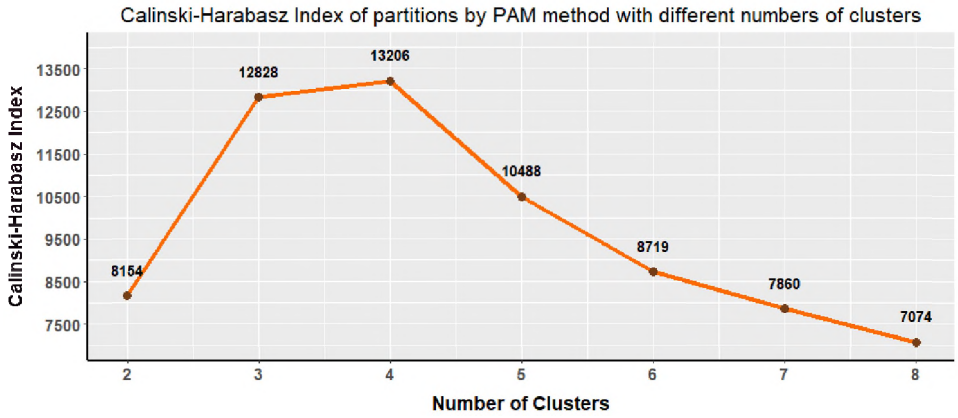
KAMILA



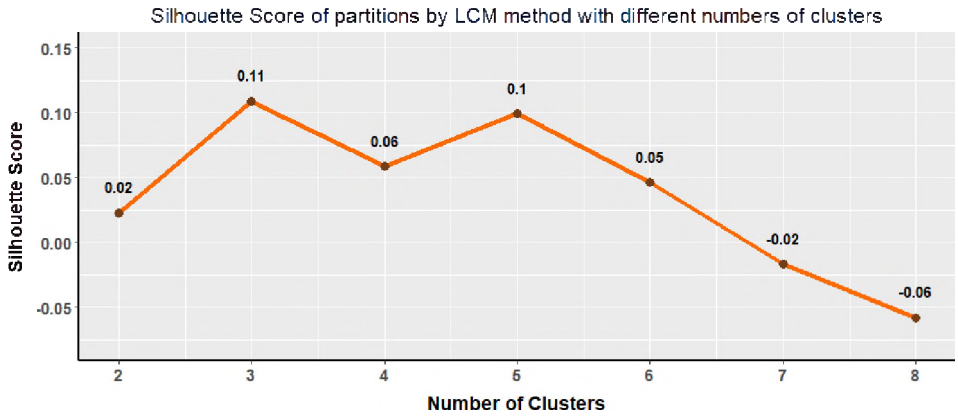


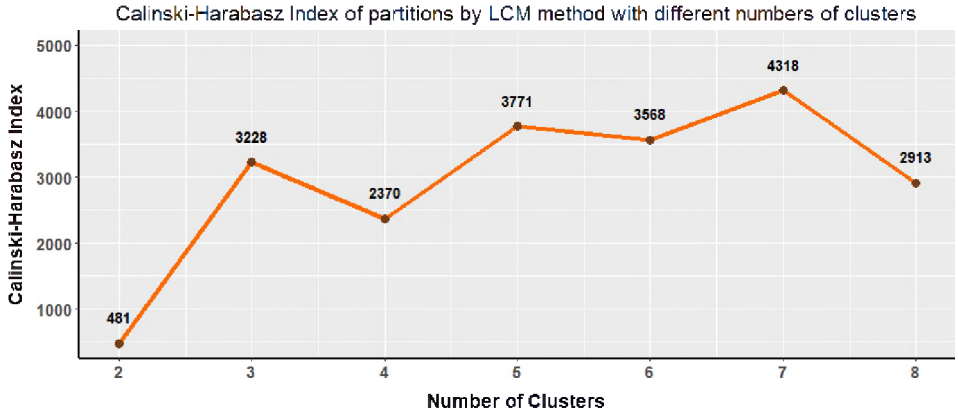
PAM


Silhouette Score of partitions by PAM method with different numbers of clusters
Silhouette Score
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.39
0.48
0.52
0.39
0.25
0.29
0.31
2
3
4
5
6
7
8
Number of Clusters

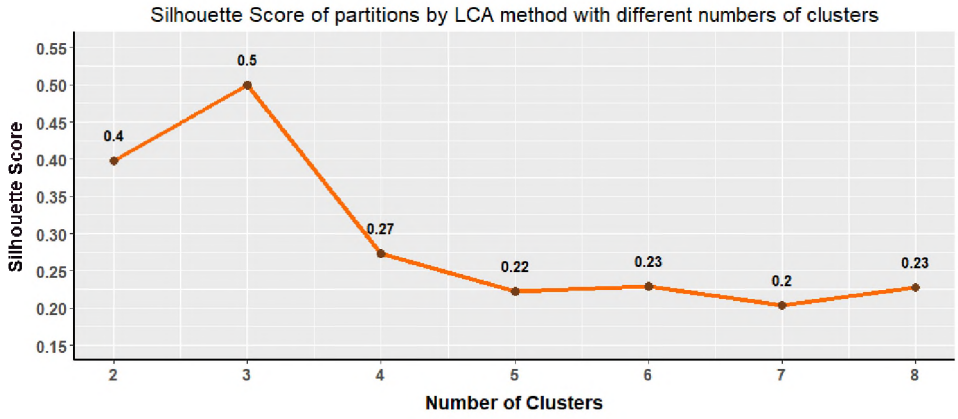

Calinski-Harabasz Index of partitions by PAM method with different numbers of clusters
Calinski-Harabasz Index
13500
12500
11500
10500
9500
8500
7500
8154
12828
13206
10488
8719
7860
7074
2
3
4
5
6
7
8
Number of Clusters

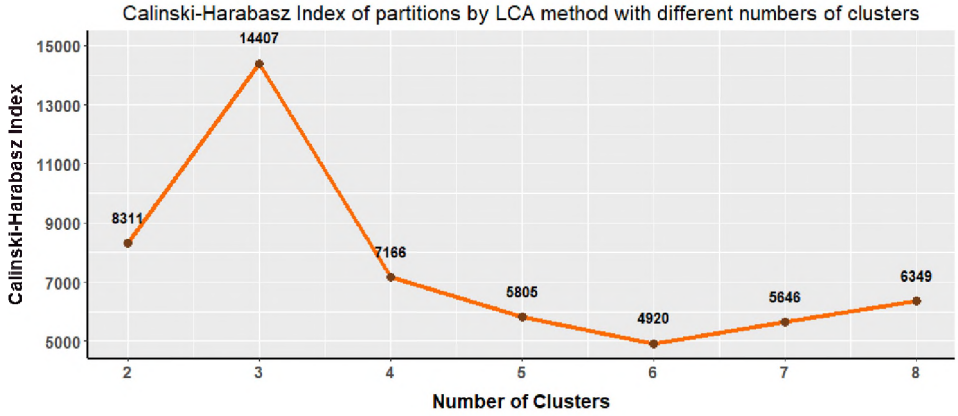
LCM


Silhouette Score of partitions by LCM method with different numbers of clusters
Silhouette Score
0.15
0.10
0.05
0.00
-0.05
0.02
0.11
0.06
0.1
0.05
-0.02
-0.06
2
3
4
5
6
7
8
Number of Clusters

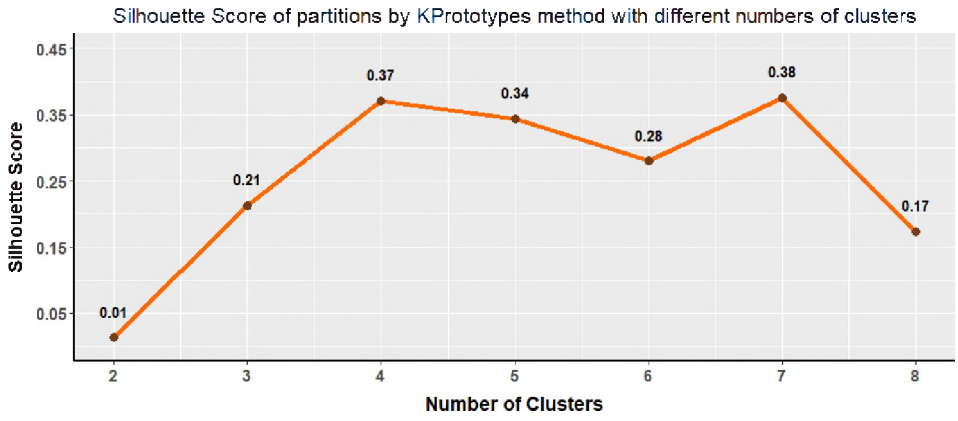

Calinski-Harabasz Index of partitions by LCM method with different numbers of clusters
Calinski-Harabasz Index
5000
4000
3000
2000
1000
481
3228
2370
3771
3568
4318
2913
2
3
4
5
6
7
8
Number of Clusters

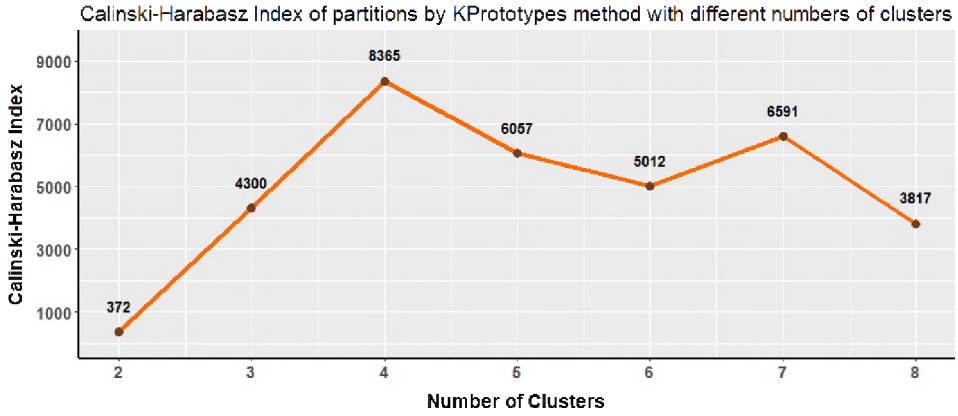
LCA


Silhouette Score of partitions by LCA method with different numbers of clusters
Silhouette Score
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.4
0.5
0.27
0.22
0.23
0.2
0.23
2
3
4
5
6
7
8
Number of Clusters

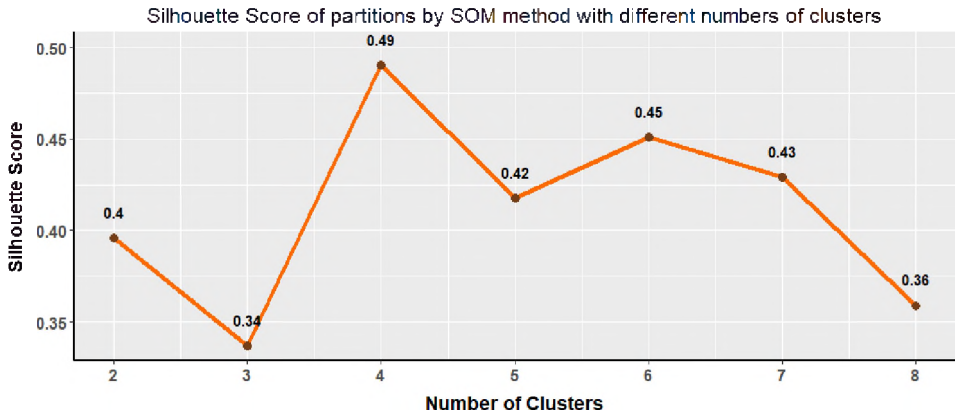

Calinski-Harabasz Index of partitions by LCA method with different numbers of clusters
Calinski-Harabasz Index
15000
13000
11000
9000
7000
5000
8311
14407
7166
5805
4920
5646
6349
2
3
4
5
6
7
8
Number of Clusters

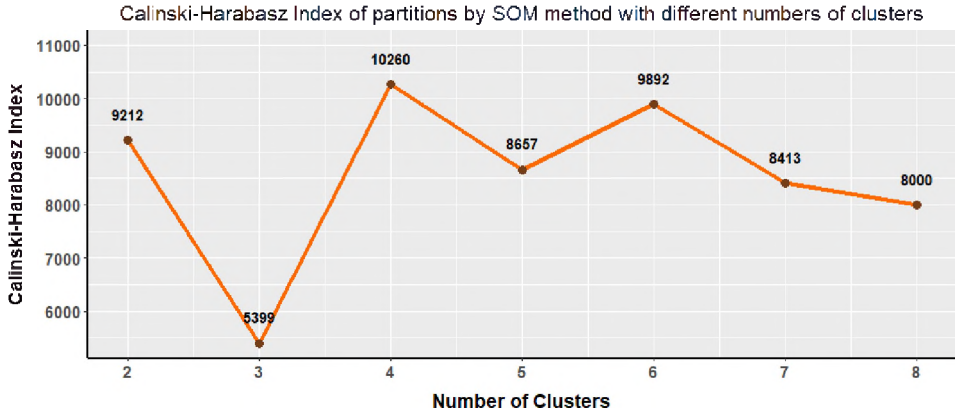
KPrototypes





SOM

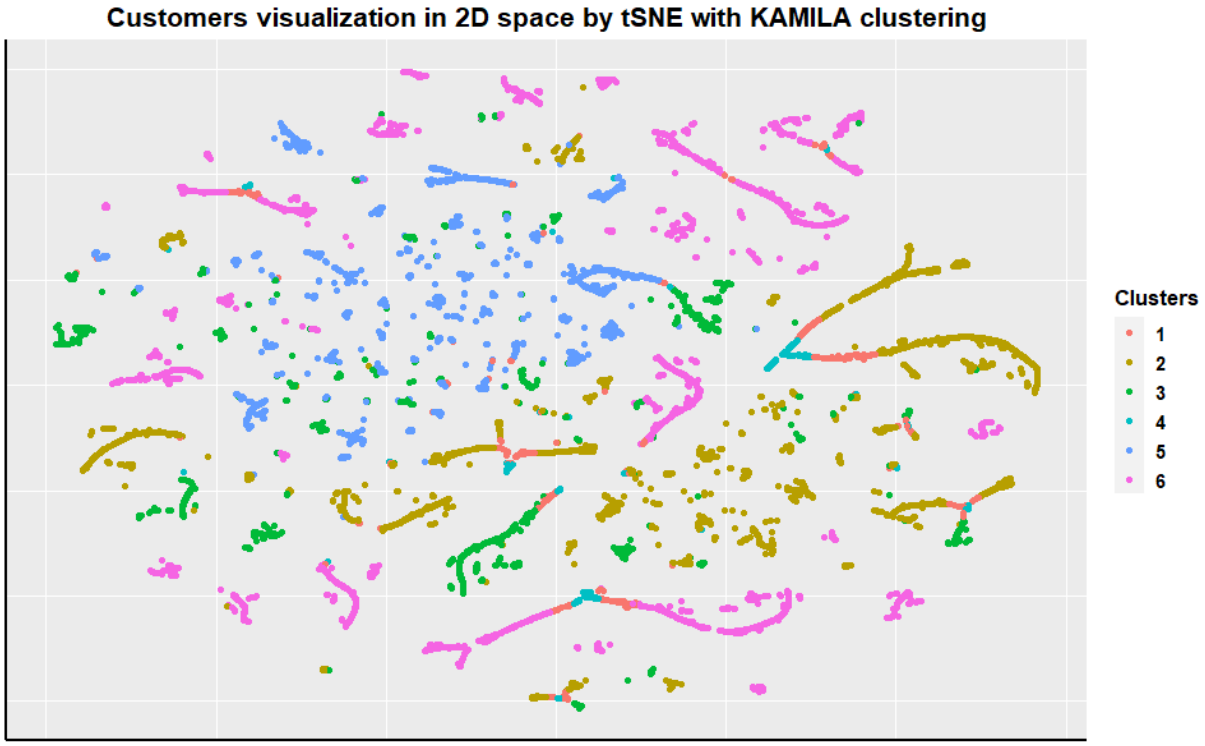

Silhouette Score of partitions by SOM method with different numbers of clusters
Silhouette Score
0.50
0.45
0.40
0.35
0.4
0.34
0.49
0.42
0.45
0.43
0.36
2
3
4
5
6
7
8
Number of Clusters


Calinski-Harabasz Index of partitions by SOM method with different numbers of clusters
Calinski-Harabasz Index
11000
10000
9000
8000
7000
6000
9212
5399
10260
8657
9892
8413
8000
2
3
4
5
6
7
8
Number of Clusters

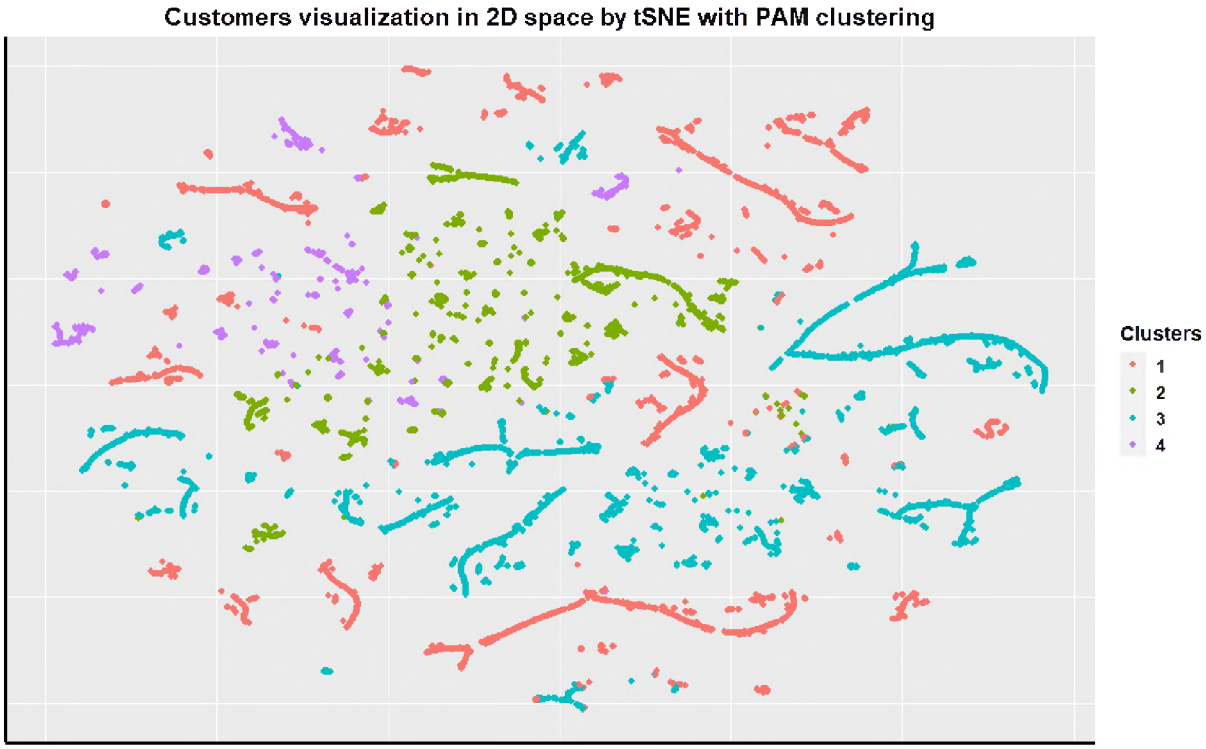
Appendix B

Clustering methods tSNE visualization

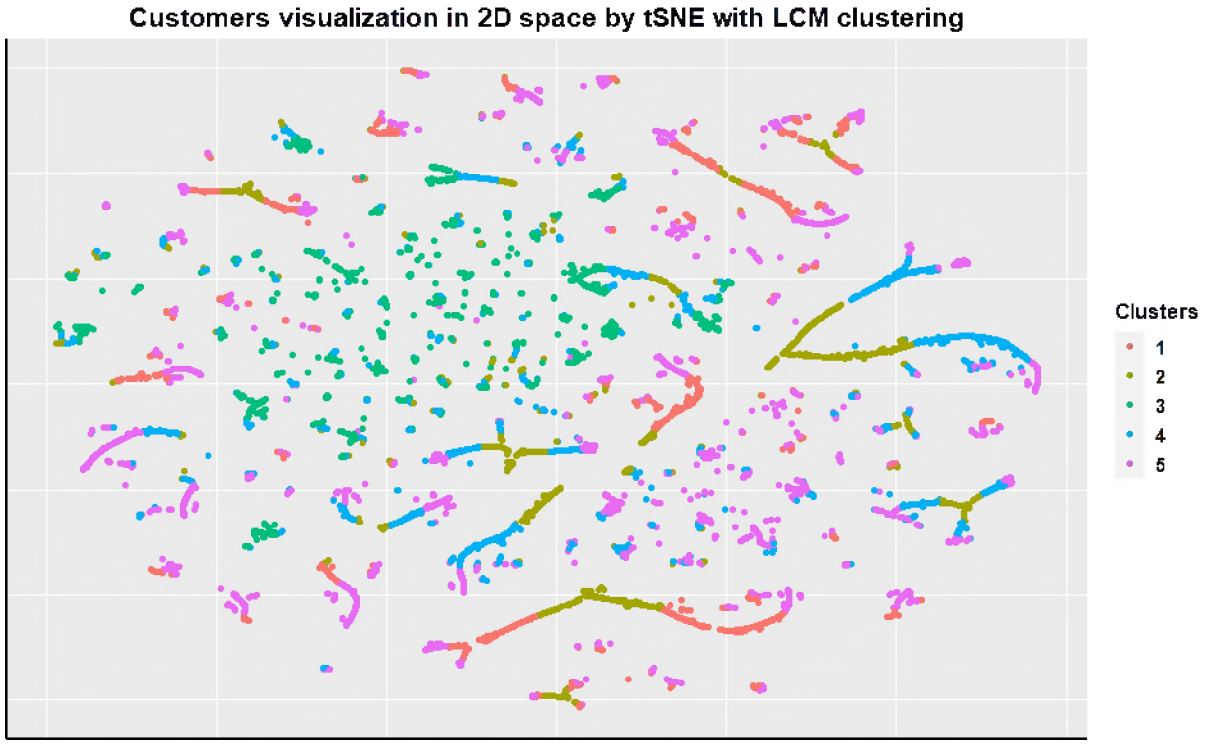
KAMILA



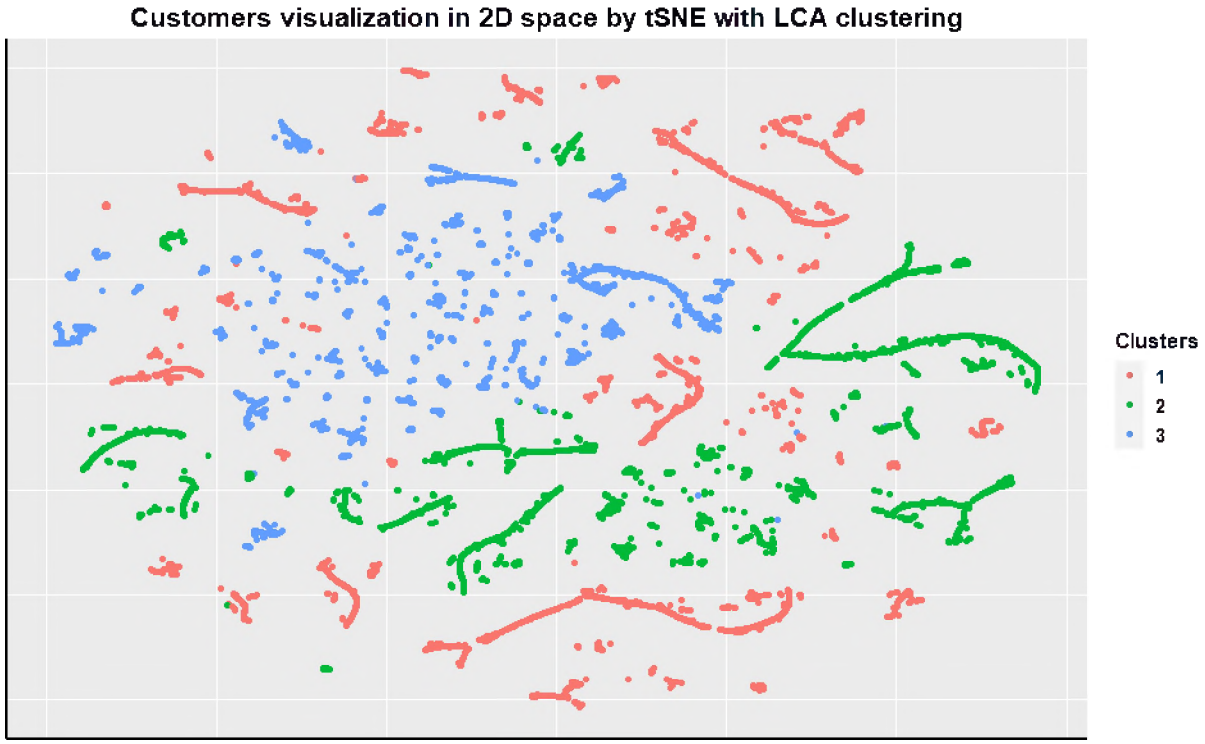
PAM


Customers visualization in 2D space by tSNE with PAM clustering
Clusters
1
2
3
4

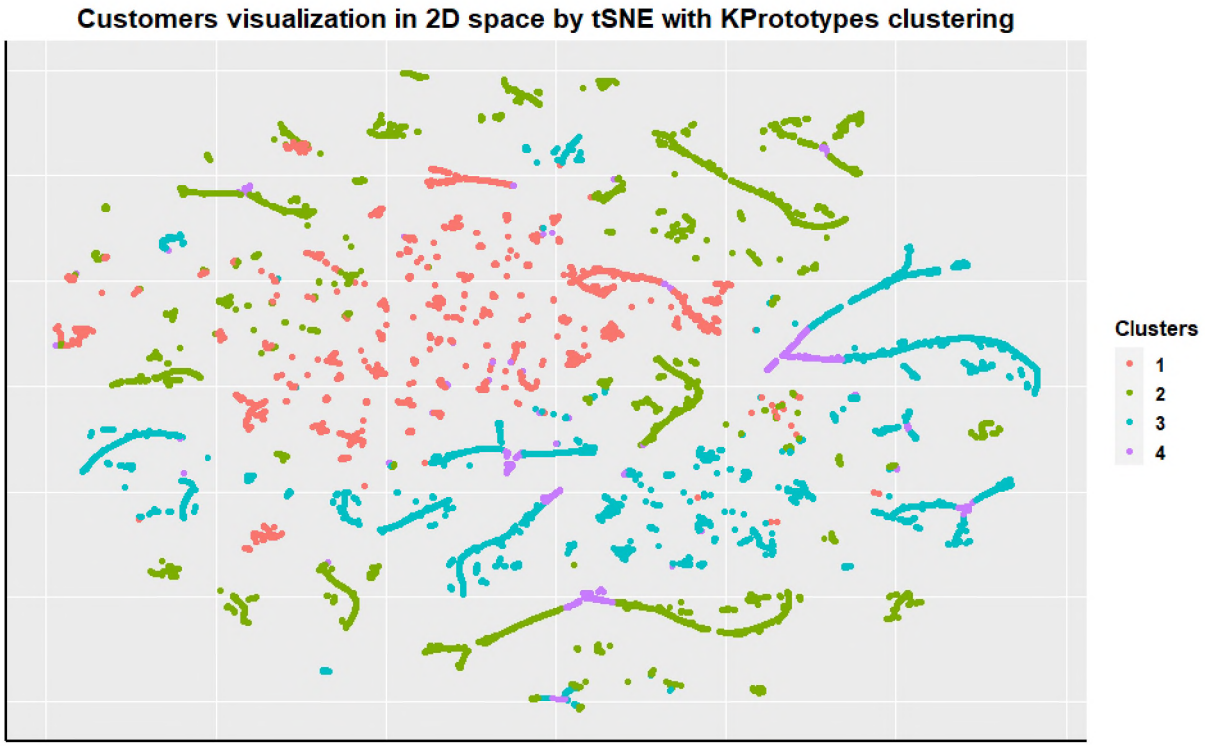
LCM


Customers visualization in 2D space by tSNE with LCM clustering
Clusters
1
2
3
4
5

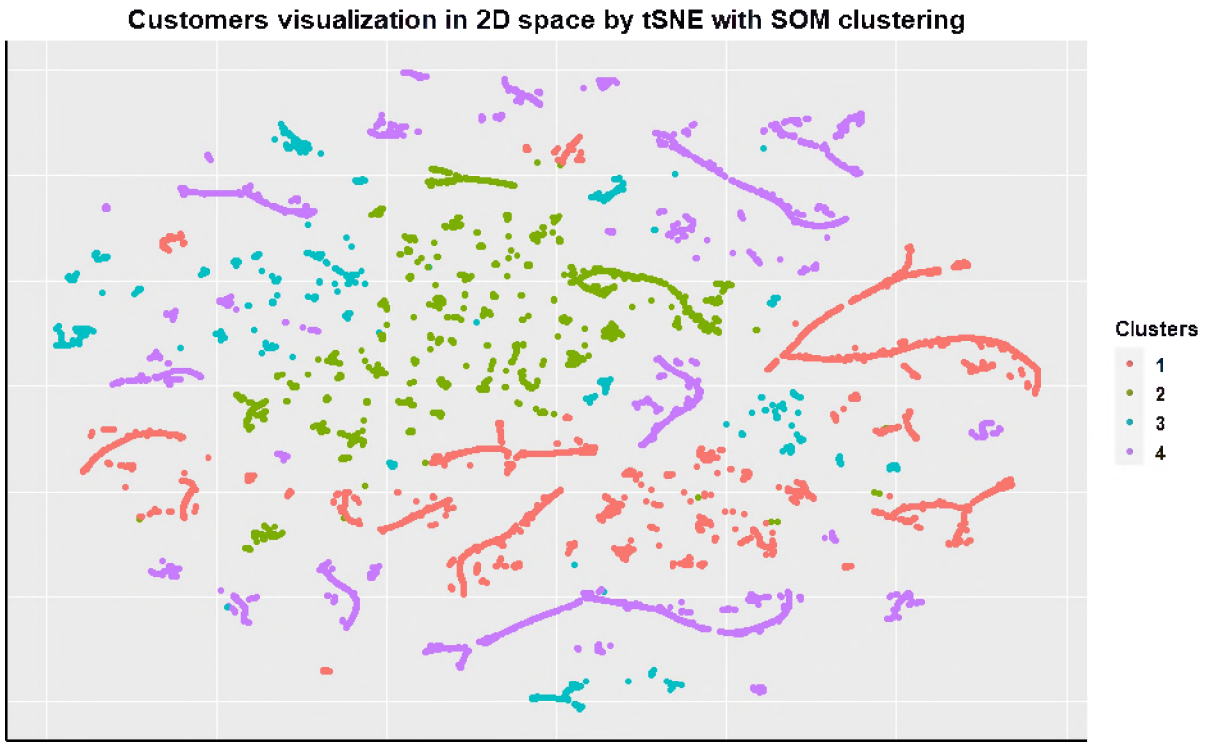
LCA


Customers visualization in 2D space by tSNE with LCA clustering
Clusters
1
2
3

KPrototypes



SOM


Customers visualization in 2D space by tSNE with SOM clustering
Clusters
1
2
3
4

Bibliography

[1] Amir Ahmad and Shehroz S. Khan. "Survey of state-of-the-art mixed data clustering algorithms". In: (2018). DOI: 10.48550/ARXIV.1811.04364. URL: https://arxiv.org/abs/1811.04364.

[2] Karen Bandeen-roche et al. "Latent Variable Regression for Multiple Discrete Outcomes". In: *Journal of the American Statistical Association* 92.440 (Dec. 1997), pp. 1375–1386. DOI: 10.1080/01621459.1997.10473658. URL: https://doi.org/10.1080/01621459.1997.10473658.

[3] Tadeusz Caliński and Harabasz JA. "A Dendrite Method for Cluster Analysis". In: *Communications in Statistics - Theory and Methods* 3 (Jan. 1974), pp. 1–27. DOI: 10.1080/03610927408827101.

[4] Clifford C. Clogg. "Latent Class Models". In: *Handbook of Statistical Modeling for the Social and Behavioral Sciences*. Springer US, 1995, pp. 311–359. DOI: 10.1007/978-1-4899-1292-3_6. URL: https://doi.org/10.1007/978-1-4899-1292-3_6.

[5] Alexander H. Foss and Marianthi Markatou. "bkamila/b: Clustering Mixed-Type Data in iR/i and bHadoop/b". In: *Journal of Statistical Software* 83.13 (2018). DOI: 10.18637/jss.v083.i13. URL: https://doi.org/10.18637/jss.v083.i13.

[6] Peter J. Green. "On Use of the Em Algorithm for Penalized Likelihood Estimation". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 52.3 (July 1990), pp. 443–452. DOI: 10.1111/j.2517-6161.1990.tb01798.x. URL: https://doi.org/10.1111/j.2517-6161.1990.tb01798.x.

[7] Zhexue Huang. In: *Data Mining and Knowledge Discovery* 2.3 (1998), pp. 283–304. DOI: 10.1023/a:1009769707641. URL: https://doi.org/10.1023/a:1009769707641.

[8] A. Jain and R. Dubes. "Algorithms for Clustering Data". In: *Prentice Hall* (1988).

[9] Matthieu Marbac and Mohammed Sedki. "VarSelLCM: an R/C package for variable selection in model-based clustering of mixed-data with missing values". In: *Bioinformatics* 35.7 (Sept. 2018). Ed. by Jonathan Wren, pp. 1255–1257. DOI: 10.1093/bioinformatics/bty786. URL: https://doi.org/10.1093/bioinformatics/bty786.

[10] Ron Wehrens and Johannes Kruisselbrink. "Flexible Self-Organizing Maps in kohonen 3.0". In: *Journal of Statistical Software* 87.7 (2018), 1–18. DOI: 10.18637/jss.v087.i07. URL: https://www.jstatsoft.org/index.php/jss/article/view/v087i07.