VILNIUS UNIVERSITY

FACULTY OF MATHEMATICS AND INFORMATICS

MODELLING AND DATA ANALYSIS MASTER'S STUDY PROGRAMME

Master's thesis

Functional Data Classification by Depth Measures. Case Study of Telecommunication Data

Funkcinių duomenų klasifikavimas pagal gylį. Telekomunikacinių duomenų atvejo analizė

Raigardas Balužis

Supervisor: prof. habil. dr. Alfredas Račkauskas

VILNIUS, 2021

Functional Data Classification by Depth Measures. Case Study of Telecommunication Data

Abstract

The thesis proposes a Functional Data Analysis approach to tackle real telecommunication industry problems. The study includes the applications of monotone smoothing, functional outliers detection and FANOVA methods. However, the main focus is concentrated towards proposing an efficient supervised functional classification method, that could be used in the telecommunication industry. Thus, multiple depth-based classification techniques are used to identify upcoming customer action based on an individual's behaviour when consuming mobile data. A comprehensive comparison of functional data classifiers is provided, and the influence of three main parameters is explored. The comparative study results demonstrate the relatively high accuracy of Maximum depth classifiers and the impact of different depth measures towards the classification results. Based on the accuracy, the Maximum depth classifier using the Fraiman and Muniz depth method emerges as the most efficient classifier in the case study.

Key words : Functional Data Analysis, Classification, Functional depth, FANOVA, Telecommunications

Funkcinių duomenų klasifikavimas pagal gylį. Telekomunikacinių duomenų atvejo analizė

Santrauka

Šiame darbe funkcinių duomenų analizės metodai naudojami spręsti telekomunikacijų srities iššūkius. Tezėje naudojami monotoninio glodinimo, funkcinių išskirčių paieškos ir FANOVA metodai. Vis dėlto, pagrindinis dėmesys sutelktas į efektyviausio funkcinių duomenų klasifikatoriaus paiešką. Todėl skirtingi klasifikavimo pagal gylį motodai yra naudojami siekant identifikuoti būsimą kliento veiksmą pagal jo mobiliųjų duomenų suvartojimą. Darbe pateikiamas detalus klasifikatorių palyginimas, o taip pat nagrinėjama trijų pagrinių parametrų įtaka klasifikatorių efektyvumui. Palyginamosios analizės rezultatai parodo, jog naudojant maksimalaus gylio klasifikavimo metodą galima pasiekti sąlyginai aukštą taiklumą, o skirtingi gylio matavimo metodai turi didelę reikšmę klasifikatoriaus efektyvumui. Atvejo analizėje, maksimalaus gylio klasifikatorius, naudojantis Fraiman ir Muniz gylį, pademonstravo geriausius rezultatus pagal klasifikavimo taiklumą.

Raktiniai žodžiai : Funkcinių duomenų analizė, klasifikavimas, funkcinis gylis, FANOVA, telekomunikacijos

Contents

1 Introduction

The telecom market in European Union countries is highly competitive. Customers can choose from multiple service providers and migrate between them effortlessly. The Universal Service Directive produced by the European Parliament and the Council ensures that all subscribers of telephone service can retain their number when migrating to another service provider (European Parliament, 2002). Thus, customers can change their operators easily without facing any difficulties or inconveniences.

Number portability is only one of the many factors which make the telecom markets sharply competitive. According to the report conducted by the European Commission, the annual churn rate in 28 EU members was about 26% in the year 2014. While the Average Revenue per User (ARPU) in the same countries was decreasing steadily and it reached approximately 16 EUR per month per user in the year 2014. This displays the maturity of the market and sets the benchmark for companies.

In the context of these trends, network operators strive to find methods that would enable them to grow the ARPU while keeping the churn rate as low as possible. Therefore, upselling and churn prevention are the key areas for improvements.

Making a customer buy something additional or more expensive can be rather challenging. Moreover offering something that users might not need can lead to customer frustration, decreased loyalty, or even churn. Thus it is highly important to be precise when offering and reach out only to those customers who are willing to buy something additional. Additionally, being able to identify the customers who are willing to churn can improve the retention process and secure a significant amount of future revenues.

Telecom companies can gather tremendous amounts of data from customers. The data stored by these companies can include call details, network data usage, and demographical data. Such data resources can be utilized to analyze the customer base and provide precise marketing as a result.

The industry is relatively old and mature data-wise. As a result, there are multiple papers and researches done that address the common telecommunication industry problems. Churn prediction is probably the most common problem. Multiple works are done to solve this problem, see Huang et al. (2012), or Amin et al. (2017). Nevertheless, very few papers use the FDA to analyze the telecommunication industry data. And even in these few papers, the FDA approach is mainly used to solve problems related to quality assurance in infrastructure. For example, Ben Slimen et al. (2017) presents an anomaly prevision model that can detect future anomalies in mobile networks by observing key performance indicators that are converted to functional data objects. Another example is the paper Aspirot et al. (2009), in which authors analyze a nonparametric regression model, where the explanatory variable is nonstationary dependent functional data, and the response variable is scalar. The suggested model is applied to estimate the quality of service for an end-to-end connection on a network.

Thus it can be said, that Functional data analysis (FDA) is a method that has not received much attention from mobile network operators yet, especially when it comes to data generated by customers' usage. Therefore in this thesis, the FDA is used to tackle telecom industry problems related to customers' behavioural patterns. This paper's main objective is to propose an efficient functional data classification method based on depth measures that would help predict an upcoming customer action. The following objectives support the main goal:

- To introduce multiple different depth measures and display the difference of them using mobile data usage dataset.
- To perform a comprehensive Functional Data Analysis on a real-world dataset.
- To compare multiple classification methods which are based on different functional depth measures and aggregation functions.

2 Functional Data Analysis

This section contains the methodological part of the thesis. At first, a short overview of the FDA is given. Secondly, a smoothing technique using a roughness penalty is introduced with an extension of monotone smoothing. After that, a brief overview of commonly used deptp methods is provided. Then a functional outlier detection method proposed by Febrero-Bande et al. (2008) is explained. Moreover, the case of analysis of variance is introduced in regard to functional data. And finally, an overview of functional supervised classification methods is given in order to address the main topic of the thesis.

2.1 Overview

Functional data analysis refers to a branch of statistical analysis that deals with random functions or surfaces that vary over the continuum. Typically, the random functions included in the sample are considered to be independent, and they represent the smooth realizations of an underlying stochastic process.

Although the main interest is in the underlying stochastic process and its characteristics, in reality, the process underneath is almost always latent. Therefore it cannot be observed directly. Data points can only be collected over time, either on a fixed or random time grid.

In the scope of this thesis, only equally spaced time grid will be considered, that is $t_j - t_{j-1} = t_{j+1} - t_j$ for all j. Nevertheless, measurements can be taken in a non-equally spaced time grid. For such case, slightly different methodology should be implied.

In most of the cases, time is the continuum over which functions are defined. In some other cases, spatial location, wavelengths, probabilities or other continuums are used.

In the FDA framework, a function is considered to be a single element of a sample. It is an atom of FDA methodology. Curve term is also commonly used as a synonym for function.

Consider the case when functional data is initially represented as a set of values $y_1, \dots, y_n$, where $y_i \in \mathbb{R}$. Also, assume that function $x(t)$ is reasonably smooth. Then the model is defined as

$$y_i = x(t_i) + \epsilon_i \tag{1}$$

where t in the continuum, y_i is the ith observation and ϵ_i is its measurement error.

Function $x(t)$ is composed of the linear combination of a set of functional building blocks $\phi_k, k = 1, \dots, K$ called basis functions. In mathematical notations, function $x(t)$ is expressed as

$$x(t) = \sum_{k=1}^{K} \alpha_k \phi_k(t) \tag{2}$$

The equivalent matrix form expression:

$$x(t) = A^T \Phi(t)$$

where $A = [\alpha_1, \dots, \alpha_K]$ is a vector of coefficients and $\Phi(t)$ is the K-vector basis functions. This expression is also called basis function expansion.

2.2 Smoothing

Smoothing is a procedure of recreating the true underlying function $x(t)$ given a finite set of measures $y_1, \ldots, y_n$, where the relationship between them is shown in the equation 1. To obtain an estimate of the function $x(t)$, one should first determine the basis functions. There are multiple basis systems that could be used to create functional data object. The common basis systems among researchers are B-spline, Fourier, wavelet, monomial and exponential basis. The first two basis functions are the most popular ones. While the Fourier basis is mostly used for periodic curves, B-splines is the go-to option when it comes to non-periodic and complex curves.

Once the basis functions are determined, curve estimation is basically reduced to the essentially multivariate parameter estimation problem of estimating parameters in matrix A.

For this case study, the B-spline basis system was used to smooth the investigated dataset. Also, the curves that are investigated are non-decreasing. Therefore a monotone smoothing technique is introduced and applied later on. Thus the next subsection is a brief introduction to the B-splines basis system and monotone smoothing procedure.

2.2.1 B-splines

A spline of order m is a piecewise polynomial of degree $m - 1$. Term "B-spline" is a short version of basis spline. B-splines are defined using three parameters: the range of validity, the knots, and the order.

Splines are constructed by dividing the interval, over which a function is defined, into L subintervals separated by a non-decreasing sequence of $\tau_l, l = 1, \ldots, L - 1$, that are called breakpoints or knots. Over each subinterval, a spline function is constructed as a polynomial of order m. Then neighbouring polynomials are joined up at the breakpoints, meaning that the functions defined over subintervals have to obtain same values at the breakpoints.

Although splines basis functions work well for many aspects, they tend to produce rather unstable fits to the data near the beginning and the end of the interval over which they are defined. This happens because there is not enough data to define them in these regions and therefore at the boundaries, function values are determined entirely by a single coefficient (Ramsay et al., 2009). This property of splines smoothing is exhibited later on in the paper using the real-world dataset.

2.2.2 Coefficients estimation

Once the functional basis is determined (in this case, it is B-splines basis), the next step is to estimate parameters in matrix A. To do that, the most common approach is the minimization of the sum of squared errors or least-squares estimation. In this case, the coefficients $A = [\alpha_1, \ldots, \alpha_K]$ are obtained by minimizing the least-squares criterion

$$SMSSE(y|\alpha) = \sum_{j=1}^{n}[y_j - \sum_{k}^{K} \alpha_k \phi_k(t_j)]^2$$

which can be expressed in such matrix form

$$SMSSE(y|A) = (Y - \Phi A)^T (Y - \Phi A)$$

where Y is a the n-vector if discrete data points.

Taking the derivative of criterion $SMSSE(y|A)$ with respect to A yields the equation (Ramsay and Silverman, 2008)

$$2\Phi\Phi^T A - 2\Phi^T Y = 0$$

Solving this for A provides the estimate $\hat{A}$ which minimizes the least squares solution

$$\hat{A} = (\Phi^T \Phi)^{-1} \Phi^T Y$$

.

This kind of approximation is suitable for the cases the residuals ϵ_j are i.i.d. with zero mean and constant variance σ^2. But in many real-world cases, this assumption can be too strong. Usually, the variance of the residuals will vary over time t. Therefore it is a common practice to use the weighted least squares criterion instead of the simple one.

The form of weighted least squares criterion is very similar, and the only difference is the weights matrix W

$$SMSSE(y|A) = (Y - \Phi A)^T W (Y - \Phi A) \tag{3}$$

where W is a symmetric positive definite matrix that allows for unequal weighting of squares and products of residuals. The weighted least squares estimate $\hat{A}$ of the coefficient matrix A is

$$\hat{A} = (\Phi^T W \Phi)^{-1} \Phi^T W Y$$

.

2.2.3 Roughness penalty

A more robust and powerful option for approximating discrete data by a function is regularization or roughness penalty. This approach brings the same benefits as the methods introduced earlier but evades some of the limitations that they contain. Also, it is a more general approach which is applicable for a wider range of functional data analysis use cases.

Firstly, let's express the "roughness" of a function as a square of the second derivative $[D^2 x(t)]^2$ of a function at time point t. This expression is also called curvature at t. Then

Ramsay and Silverman (2010) offers to use the integrated squared second derivative as a measure of function's roughness

$$PEN_2(x) = \int [D^2 x(s)]^2 ds$$

Having this, the penalized sum of squared errors fitting criterion can be introduced. It is a slight modification of least-squares fitting criterion 3 and is defined as the penalized residual sum of squares

$$PENSSE_\lambda(x|Y) = [Y - x(t)]^T W [Y - x(t)]^2 + \lambda \times PEN_2(x) \tag{4}$$

where λ is a smoothing parameter that controls the penalty that is placed on the roughness of a curve. Therefore a small λ makes the curve more variable and vice-versa.

Having this, the expression for the estimated coefficient matrix can be obtained

$$\hat{A} = (\Phi^T W \Phi + \lambda R)^{-1} \Phi^T W Y$$

, where

$$R = \int D^m \phi D^m \phi^T$$

The full explanation of how such expression is obtained can be found in Ramsay and Silverman (2010).

2.2.4 Monotone smoothing

Often functions that are estimated have to satisfy some side constraints. For example, estimating growth curves implies that negative slopes should not be observed. In the case study of telecommunication data, cumulative measures are used. Therefore the smoothed curves are expected to be monotonically increasing. In such cases, the monotone smoothing technique might be used.

Assume that $w(t)$ is a curve that has one constrain $w(t_0) = 0$. When $\exp[w(t)]$ is a positive function. And the indefinite integral of a positive function $\exp[w(t)]$ is always increasing. As a result, a monotone smoothing function can be expressed in such form (Ramsay et al., 2009):

$$y_j = \beta_1 \int_{t_0}^{t_j} \exp[w(u)] du + \epsilon_j$$

, where t_0 is the fixed origin for the range of t-values for which the data are being fit.

or in the matrix form:

$$y_j = \beta_1 \int_{t_0}^{t_j} \exp[\Phi(u)^T A] du + \epsilon_j$$

.

2.3 Functional Data Centrality measures

This subsection is dedicated to a brief overview of the location estimators of the center curve. The main focus is dedicated towards the functional depth (FD), and the different methods of its calculation.

The key tool in describing the centrality or the outlyingness of a curve in the FDA is depth. The notion of functional data depth allows the ordering of the functional data objects. A curve x_i with the highest functional depth $FD(x_i)$ corresponds to being center curve in the set, while the lowest functional depth represents the opposite - farthest curve from the center.

There are multiple methods for depth estimation. In the following subsections, different notions of data depth are introduced. All of the described depth functions are later being used in the classification models for mobile data usage curves.

2.3.1 The Fraiman and Muniz (FM) method

The concept was created by Fraiman and Muniz (2001), and it is defined below.

For each $t \in [0,1]$ let

$$F_{n,t}(a) = \frac{1}{n}\sum_{k=1}^{n} 1_{\{X_k(t)\leq a\}}$$

be the empirical distribution of the sample $X_1(t), X_2(t), \ldots, X_n(t)$ and let $Z_i(t)$ denote the univariate depth of the data $X_i(t)$ in this sample given by

$$D_i(t) = 1 - |\frac{1}{2} - F_{n,t}(X_i(t))|$$

Then, define for $i = 1, \ldots, n$,

$$I_i = \int_0^1 D_i(t)dt$$

and rank the observations $X_i(t)$ according to the values of I_i, $i = 1, \ldots, n$.

2.3.2 *h*-mode (hM) method

The h-mode depth method was proposed as a functional generalization of the likelihood depth in order to quantify how surrounded one function is by the others. For a datum x, the population hM depth is given by

$$f_h(x_0) = (E)[K(m(x_0, X)/h)]$$

where X is a random element, which describes the population, m is a suitable metric, $K(\cdot)$ is a kernel and h is the bandwidth parameter (Cuesta-Albertos et al., 2016).

2.3.3 Random Projections method

This method was first introduced by Cuevas et al. (2007).

Given a sample of curves $X_1(t), X_2(t), \dots, X_n(t)$ they are projected on a random direction a (independent from $X_i(t)$) usually by calculating the inner product $\langle a, X\rangle = \int_0^1 a(t)X(t)dt$. The sample depth of a datum X is the univariate depth of the projection $\langle a, X\rangle$ with respect to the projected sample $\{\langle a, X\rangle\}_{i=1}^{N}$.

Usually, several directions $a_1, \dots, a_R, R > 1$ are generated and the projections can be summarized using different methods. In this case, mean is used to aggregate the multiple depths into a single representative number and a is chosen according to Gaussian distribution as it was proposed originally by Cuevas et al. (2007). So if $D_{a_r}(x)$ is the depth of the rth projection, then

$$RPD(x) = R^{-1}\sum_{r=1}^{R} D_{a_r}(x)$$

2.3.4 Double Random Projections method

This method is very similar to the Random Projections method, and it was also introduced by Cuevas et al. (2007).

Assume that $X_1(t), X_2(t), \dots, X_n(t)$ is a sample of differentiatable functions defined on $[0, 1]$. Then instead of using only the initial functions as it is done in Random Projections method, derivatives of the functions are also incorporated. This brings the information on the function smoothness. In mathematical terms, the sample of functions $X_1(t), X_2(t), \dots, X_n(t)$ are reduced to a sample of pairs of inner products between the random direction a and function and its derivative $(\langle a, X_1\rangle, \langle a, X_1'\rangle), \dots, (\langle a, X_n\rangle, \langle a, X_n'\rangle)$. This bi-dimensional sample is could then be aggregated in different ways. But in the scope of this work, h-mode depth is used, which is also introduced in Cuevas et al. (2007).

2.4 Outlier detection

Outlier detection is a crucial step of any statistical analysis. Different methods might be used to detect the abnormal records; therefore, this subsection is a short introduction about the method that is used in this case study.

In the FDA framework, a curve is considered an outlier if generated by a process with a different distribution than the rest of the identically distributed curves (Febrero-Bande et al., 2008).

In order to detect outliers in the functional dataset, the notion of depth is used. The functional depth and outlyingness are inverse terms. Therefore a curve can be considered to be an outlier if it has significantly lower depth compared to the majority of the curves. Based on this fact, depth is used as the main characteristic in the outlier detection procedure. For this particular case, RP depth is used.

The procedure of functional outlier detection, proposed by Febrero-Bande et al. (2008) consists of 3 main steps:

1. Obtaining functional depths of each curve in every group separately;
2. Removing the curves for which the depth $D(x_i) \leq C$ for a given cutoff value C;
3. Come back to the first step with the reduced dataset after step 2 and repeat this procedure until no more outliers are detected.

The key point of the described algorithm is selecting a proper cutoff C. Ensuring a reasonable level of type I errors is the main criteria when determining the value of C. As proposed by the authors of the algorithm, such C is selected, that the proportion of non-outlier observations labelled as outliers (false positives) would be approximately equal to 0.01 or (1%). In mathematical terms, such C is selected, that in the absence of outliers

$$P(D_n(x_i) \leq C) = 0.01$$

where $i = 1, \dots n$.

Therefore, the estimated 1th percentile of the observed sample curves is taken as C. As there might be multiple outliers in the sample, a robust estimate of the percentile has to be obtained. To do that, two bootstrap procedures might be used. The two procedures differ in one step only, which defines the sample selection before the bootstrap. The first one is based on trimming the sample by removing possible outliers, while the second method is based on bootstrapping the curves of the original dataset with probability proportional to their depth.

In the case study, the trimming procedure is used; therefore, the algorithm for it is explained below.

1. Obtaining functional depths of each curve in every group separately;
2. Trimming the dataset of curves by deleting the $\alpha\%$ less deepest curves;
3. Obtaining the the B standard bootstrap samples of size n from the trimed dataset;
4. For each bootstrap set $b = 1, \dots, B$, obtaining a cutoff C^b as an empirical 1% percentile of the distribution of the selected depths $D(y_i^b)$, where $i = 1, \dots, n$;
5. Taking C as the median of all obtained cutoffs C^b, where $b = 1, \dots, B$.

This algorithm is later being applied in the case study for the telecommunication data.

2.5 FANOVA

This subsection is dedicated to a short introduction to the analysis of variance test for functional data.

Assume that there are a l groups of independent random functions $X_{ij}(t), i = 1, \ldots, l, j = 1, \ldots, n_i$ where $t \in [0, 1]$ and $n = n_1 + \cdots + n_l$. $\mu_i(t)$ denotes the mean function, while $\gamma_i(s, t)$ denotes the covariance function for each group of functions $i = 1, \ldots, l$.

Then the null hypothesis is such:

$$H_0 : \mu_1(t) = \cdots = \mu_l(t), t \in [0, 1] \tag{5}$$

.

While the alternative hypothesis is the negation of the H_0 - there are at least two groups of functions with non-equal mean functions μ_i.

Testing such a hypothesis is called one-way analysis of the variance problem for functional data (FANOVA). There are multiple tests created for testing the hypothesis 5. The tests can be split into two big groups: the ones which are based on the F test statistic and the other group of tests is based on L^2-norm-based tests.

A short introduction to these groups of tests is provided. A comprehensive comparison of the FANOVA tests is made by Górecki and Smaga (2015).

The first group of tests are based on pointwise F test statistic (Ramsay and Silverman, 2010) which is expressed in such formula:

$$F_n(t) = \frac{SSR_n(t)/(l-1)}{SSE_n(t)/(n-l)} \tag{6}$$

where

$$SSR_n(t) = \sum_{i=1}^{l} n_i(\bar{X}_i(t) - \bar{(}X)(t))^2, \tag{7}$$

is the pointwise between-subject variation,

$$SSE_n(t) = \sum_{i=1}^{l}\sum_{j=1}^{n_i}(X_{ij}(t) - \bar{X}_i(t))^2 \tag{8}$$

is the pointwise within-subject variation, $\bar{X}(t) = \frac{1}{n}\sum_{i=1}^{l}\sum_{j=1}^{n_i} X_{ij}(t)$ is the sample grand mean function and $\bar{X}_i(t) = \frac{1}{n_i}\sum_{j=1}^{n_i} X_{ij}(t), i = 1, \ldots, l$ is the sample group mean function.

The second group of FANOVA tests are the modifications of L^2-norm-based tests proposed by Cuevas et al. (2004).

In their work,

$$\sum_{1 \le i < j \le l} n_i \int_I \bar{X}_i(t) - \bar{X}_j(t))^2 dt$$

was considered as a test statistic.

In the part of the analysis of this paper, several tests from each group are used to check the null hypothesis.

2.6 Classification

Clustering and classification are helpful tools in both traditional data analysis and functional data analysis. Clustering is a procedure of grouping a set of data into such an arrangement that data objects within the same clusters are more similar than across clusters with respect to a particular metric. Meanwhile, classification aims to assign an individual observation to a pre-determined group based on labelled observations. Using the machine learning terminology, functional data clustering is an unsupervised learning process while functional data classification is a supervised learning procedure (Wang et al., 2015). Clustering aims to construct groups based on some clustering criteria, while classification defines the class of a new data object using a discriminant function or a classifier.

In this thesis, only supervised classification approach is used. Thus it is discussed more broadly.

2.6.1 Functional Supervised Classification

The aim of supervised classification is to assign group membership to a new data object using a particular classifier. The standard framework for supervised classification has been mainly developed for the case when the predictor variables X, Y are $\mathbb{R}^d$-valued, with small values of the dimension d, while the functional classification methods are capable of dealing with functional X and Y defined over a continuous interval, usually $[0, 1]$.

Depending on the number of groups k present in the dataset, the classification can be separated into two types: binary ($k = 2$) and multi-class ($k > 2$) classification. In this section, binary classification algorithms are discussed, as this methodology is later being used in the case study.

The supervised classification task for two groups (sometimes called populations) can be described as such: assume that there are two independent training samples $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$ taken from the populations P_0 and P_1 that consist of random variables X and Y. A classifier's goal is to assign a new observation Z to either P_0 or P_1 based on the information given by the training samples.

In the FDA framework, variables X, Y, Z are functional data objects. Therefore the classifier has to, first, "learn" from the labelled curves and afterwards use that information when an unlabeled function is introduced.

Supervised classification procedure usually includes the following steps:

1. Splitting the data into two parts: train and test sets.
2. Training the chosen classifier using the train set. The result of this step is a classification model which is used to predict the label of a new observation.

3. Using the pre-trained classifier to label the records from the test set.

4. Measuring classifier performance via selected performance measures.

In the thesis, classifiers that are based on depth measures are used and compared.

The Maximum depth classifier was the firstly introduced by Liu (1990), while Ghosh and Chaudhuri (2005) fully develop this procedure. It was the original attempt to use data depths instead of multivariate raw data in supervised classification problems.

The idea behind this classifier is fairly simple. It classifies an observation to a class to which observation has the maximum depth. In mathematical terms, the classifier can be represented as

$$d_D(x) = \arg\max_j D_{n_j}(j, x) \tag{9}$$

where n_j is the number of training set observations from the jth population, D_{n_j} is the empirical depth of a function x in the jth population ($j = 1, 2$) and the prior probabilities of the competing classes are assumed to be equal (Ghosh and Chaudhuri, 2005).

Note that any depth functional measure can be used to determine the depth of observation, and it can be considered as a parameter of a model.

While maximum depth classifier might work well on some occasions, it is a linear classifier, and it might perform poorly in some cases.

In order to better understand when maximum depth classifier fails to perform well, a DD-plot notion is used. The *depth vs. depth plots* were first introduced by Liu et al. (1999) as a graphical comparison of two multivariate samples. Precisely, a DD-plot is a plot of $DD(F, G)$, where

$$DD(F, G) = \{(D_F(x), D_G(x)), \text{for all } x \in \mathbb{R}^k\}.$$

Here F and G are two continuous distributions in $\mathbb{R}^k$ (Li and Liu, 2004). And the output of a DD-plot is a two-dimensional scatterplot, which contains depth measures of different groups on both axes.

It is important to notice that $DD(F, G)$ always belongs to $\mathbb{R}^2$ no matter how large is the dimension k. This property serves well when analyzing functional data objects.

Coming back to the drawbacks of maximum depth classifier, an example provided by J.A. et al. (2017) displays a case when it succeeds and fails to produce an appropriate result (see Figure 1). The diagonal is added to the figures as a maximum depths classification rule. And in the first case it does the job fairly well (DD-plot #1), but in the second case (DD-plot #2) it fails when classifying almost all observations to group Q.

This drawback displays the need for a more flexible classifier. For this reason, Li et al. (2012) proposed replacing the diagonal rule (maximum depth) by a function, such that it

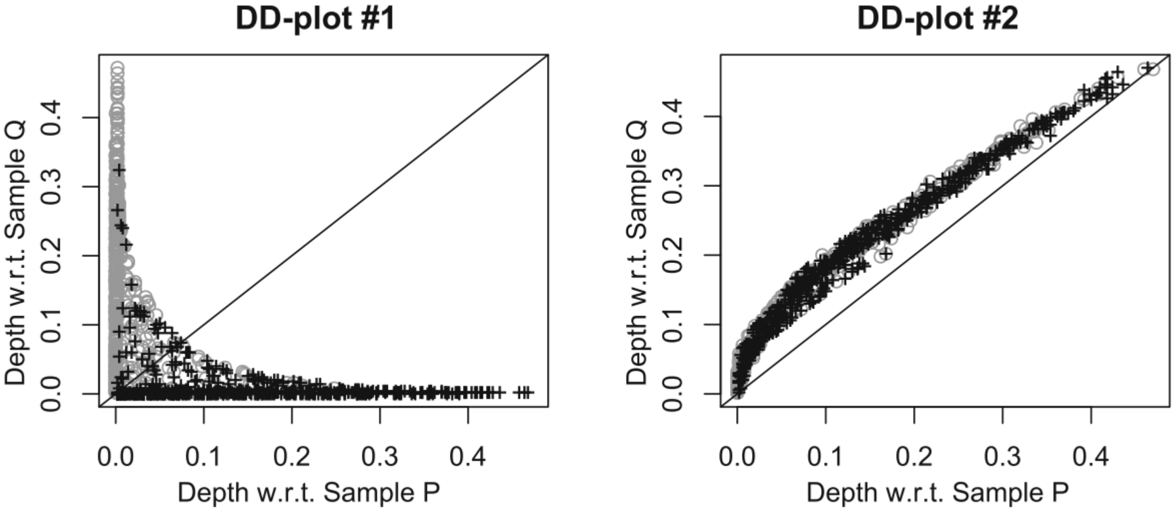


Figure 1: DDplot takenFrom article Liu2004.
+ and o mark observations belonging to different groups.

splits the DD-plot into two areas, which achieves the lowest misclassification rate. This approach is called DD-classifier.

DD-classifier is a non-parametric classifier, which aims to find a curve which best separates the two samples in their DD-plot by minimizing the classification error. It has a few advantages named by Li et al. (2012), which are the following:

- It is completely data-driven, and the best separating curve is determined by the underlying probabilistic geometry of the data.
- Since the classifier is based on depth, it implies the standardization effect. As a result, estimating parameters such as means and scales are not required.
- The output of classification can be visualized well using a two-dimensional DD-plot.

The following description is again provided for a two-class case.

Assume that $\{X_1, \ldots, X_m\}$ and $\{Y_1, \ldots, Y_n\}$ are two samples from F and G which are continous distributions in $\mathbb{R}^k$. The first step in estimating the best separating function $r_0(\cdot)$ is the selection of a particular family of functions Γ. Then given any $r \in \Gamma$, such classification algorithm is considered:

$$\begin{cases} \text{if } D_{G_n}(x) > r(D_{F_m}(x)) & \text{assign } x \text{ to } G \\ \text{if } D_{G_n}(x) \leq r(D_{F_m}(x)) & \text{assign } x \text{ to } F, \end{cases} \tag{10}$$

where $D_F(x) = l_1(f_1(x))$ and $D_G(x) = l_2(f_2(x))$. Here $f_1(\cdot)$ and $f_2(\cdot)$ are both from the elliptical family and l_1, l_2 are some strictly increasing functions.

And the aim here is to find an optimal $r_0 \in \Gamma$ from a prefixed Γ, that would minimize the overall misclassification rate.

So for any $r \in \Gamma$, a curve $y = r(x)$ is drawn on the DD-plot. It determines, that the observations above the curve belong to class G while the observations below belong to F. Having the points classified, the empirical missclassification rate is calculated using such formula:

$$\hat{\Delta}_N(r) = \frac{\pi_1}{m}\sum_{i=1}^{m} I_{\{D_{G_n}(X_i) > r(D_{F_m}(X_i))\}} + \frac{\pi_2}{n}\sum_{i=1}^{n} I_{\{D_{G_n}(Y_i) \leq r(D_{F_m}(Y_i))\}}, \tag{11}$$

where π_1 and π_2 are the prior probabilities of F and G respectively, $N = (m, n)$ and $I_{\{A\}}$ is an indicator function that takes 1 if A is true and 0 otherwise.

As a result, Li et al. (2012) propose to estimate the optimal r_0 by $\hat{r}_N$, which minimizes the empirical misclassification rate $\hat{\Delta}_N(r)$. If $\hat{r}_N = \arg\min_{r\in\Gamma} \hat{\Delta}_N(r)$, the proposed DD-classifier is

$$\begin{cases} \text{if } D_{G_n}(x) > \hat{r}(D_{F_m}(x)) & \text{assign } x \text{ to } G \\ \text{if } D_{G_n}(x) \leq \hat{r}(D_{F_m}(x)) & \text{assign } x \text{ to } F. \end{cases} \tag{12}$$

Other nonparametric classifiers can also be applied in order to predict the groups using the two-dimensional DD-plot. Methods like Linear and Quadratic discriminant analysis (LDA, QDA), k-Nearest Neighbour (KNN) classification is used as a comparison to maximum depth (MD), and DD-classifier approaches.

3 Data

Data used for the case study is real, and it comes from one of the telecommunication service providers within the European Union member state.

The dataset contains information about 652 SIM cards and the way they were used to consume mobile data throughout the period of 1 year.

For each SIM card, there are 365 records that represent 12 months of mobile data usage starting from 1st of September 2019 up until 1st of September 2020. One observation represents the amount of mobile data a single customer used throughout the day. Daily mobile data usage is measured in megabytes.

Additionally, the dataset includes a categorical variable which is called "Action" throughout this work. This variable represents the action made by the SIM card holder on the next (13th) month. There are four possible values that this variable might have. These values cover the whole set of alternatives. The actions are such:

- *Upgrade* - monthly data allowance is increased;
- *Downgrade* - monthly data allowance is decreased;
- *Churn* - a customer stops using the company's service;
- *No action* - none of the above actions are performed.

Table 1 contains the number of customers each category covers.

Table 1: Number of customers in each category of Action variable

Action	**Number of customers**
Upgrade	317
Downgrade	93
Churn	104
No action	138

This variable is later being used as the main target variable in the classification task.

3.1 Data preparation

Before performing the functional data analysis, several data preparation steps were executed. This section provides the motivation and explanation of these steps.

At first, transformations were made to the initial dataset. Values of daily mobile data usage were express in proportional cumulative daily sums. First of all, daily data usage measures were converted into proportions of monthly data allowance. Then the proportions were summed cumulatively. An example of this transformations is displayed in table [2]

Table 2: Example of mobile data usage transformation for a customer with 10GB monthly allowance

Day	Usage (MB)	Proportion	Cumulative proportion
1	500	0,05	0,05
2	550	0,055	0,105
3	250	0,025	0,13
4	400	0,04	0,17
...	...	...	...
31	250	0,025	0.9

Figure 2 illustrates the data used and the main goal of the analysis - estimating the action made during September of 2020. The prediction of the upcoming action is based on a customer's historical behaviour.

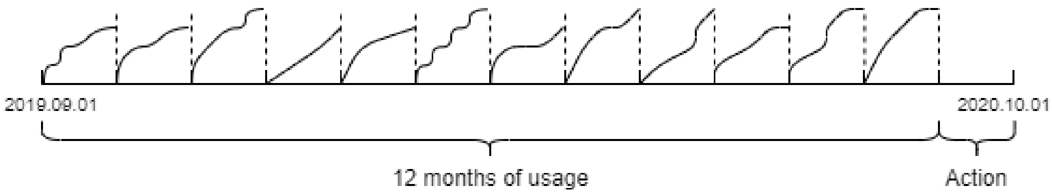


Figure 2: Proportional cumulative sums of daily mobile data usage

3.2 From discrete data to functional data

Step zero in functional data analysis is creating functional data objects. In order to do that, discrete data points are smoothed using basis functions. Since the investigated dataset contains cumulative sums, monotone smoothing method is applied.

At first, every month of each customer data usage is smoothed using B-splines basis. This results in 12 curves for each customer. A result of such procedure is displayed in the figure 3, where one curve is fitted on the data points of one month. The lower graph of figure 3 represents the residuals of the smoothing procedure.

Different months have different lengths and smoothing parameters depend on the number of time points in one curve. Therefore three separate smoothing procedures were executed according to the month length. Smoothing parameters are displayed in the table [3].

A breakpoint at each time point (day) was created. As done in the large majority of applications, only a single knot is used at every breakpoint. Although there are several methods for obtaining optimal parameter λ value, the goal of the case study is related to classification. Therefore this parameter was obtained by trial and error, taking into account the residuals of the fitted curve and analyzing the smoothed curves visually.

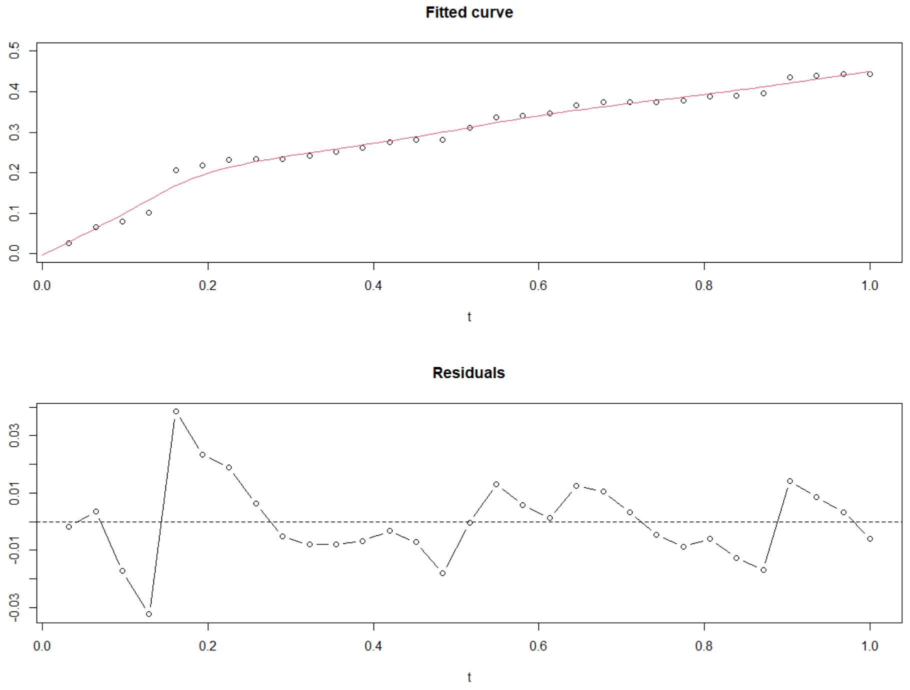


Figure 3: Curve of single customer for month 2020.08

Table 3: Smooting parameters used for different months

Months	Days in month	# of interior knots	# of basis functions	Order of B-splines	λ
2020.02	29	27	30	3	0.000005
2019.09 2019.11 2020.04 2020.06	30	28	31	3	0.000005
2019.10 2019.12 2020.01 2020.03 2020.05 2020.07 2020.08	31	29	32	3	0.000005

Combinations of smoothing parameters presented in table 3 are sufficient to capture customer behaviour's primary trend while reducing the impact of random "high jumps" in day-to-day customer usage.

An example of smoothed months for one of the customers is provided in figure 4.

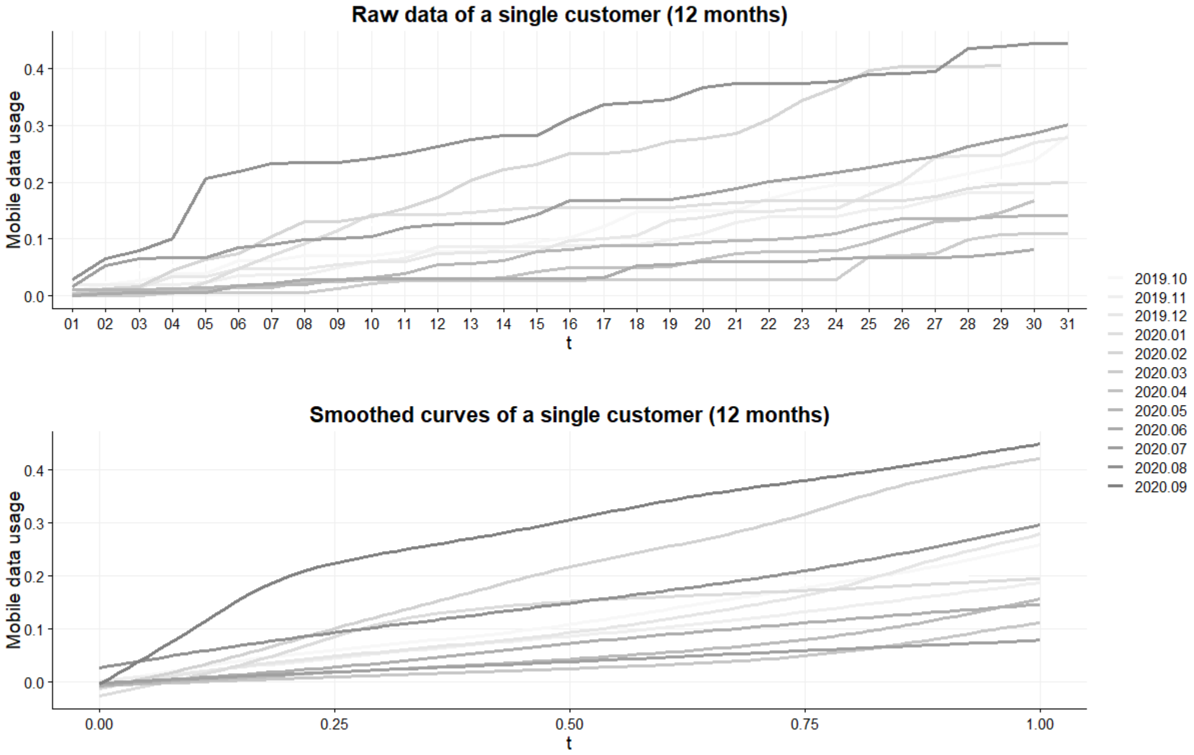


Figure 4: Smoothed curves of single customer (12 months)

3.2.1 Aggregation using weighted mean

Once monthly curves are created for each customer, aggregations of these curves are applied in order to have a single curve associated with a single customer. The weighted average is used to aggregate the curves on the customer level. Different combinations of weights are tested in order to receive the best classification results.

The weights for months are generated using the function

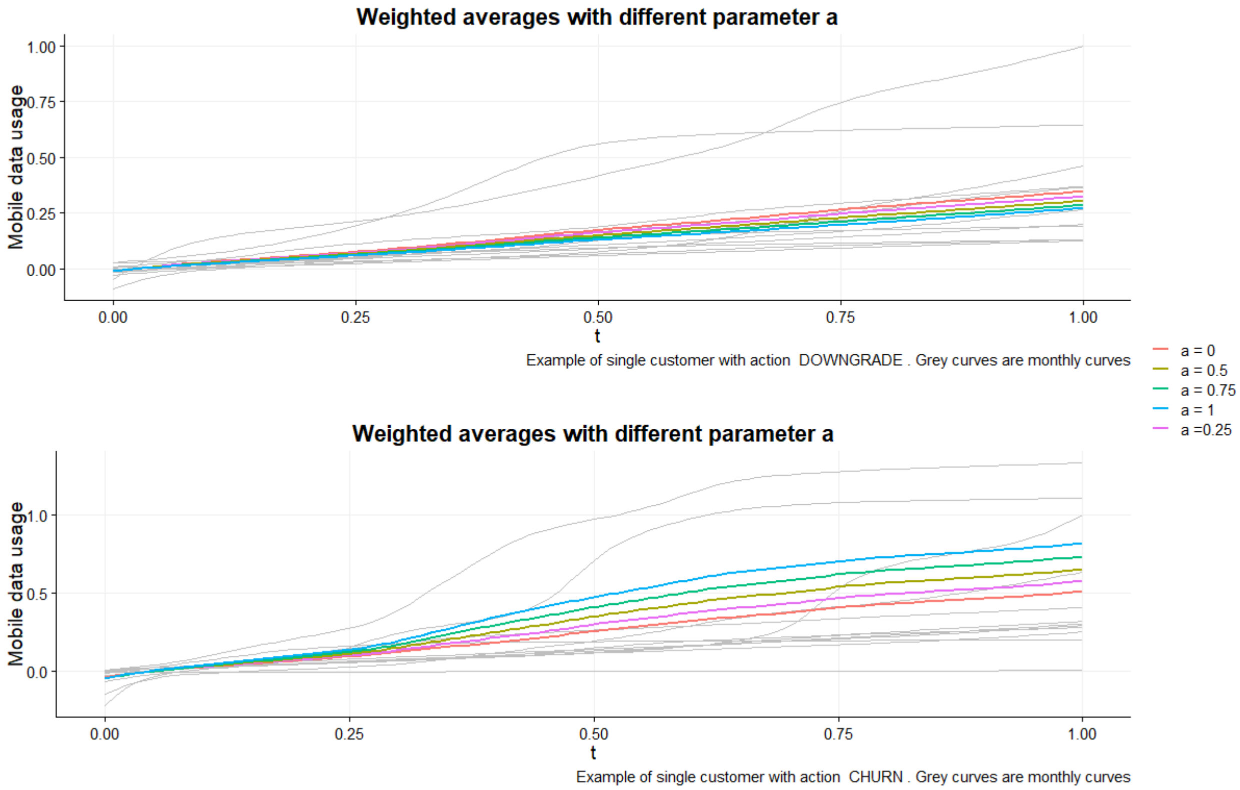
$$w_k = \frac{1}{k^a},$$

where k_i is a month index, such that $k_{2019.10} = 12, k_{2019.11} = 11, \dots, k_{2020.09} = 1$ and $a \in \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$.

When $a = 0$, all months get the same weight, meaning that no behavioural shift is assumed in the monthly data usage. In contrast, when $a > 0$, bigger weights are assigned to more recent months. This implies that throughout the year, customer behaviour changes and the data from recent months might bring more information about what will happen in the upcoming month.

Two customers from groups *Downgrade* and *Churn* are selected. As seen from the upper plot of figure 5, customer's data usage dropped in more recent months (before downgrading the plan). As a result, the blue curve, which marks the weighted mean with $a = 1$, is slightly lower compared to others.

As for the lower plot in the figure 5, the customer started using more data in the recent months, but in the majority of older months, the usage was lower (not even reaching 0.5 at the end of the month). Nevertheless, the weighted mean with $a = 1$ is significantly higher than other mean functions, as it puts more importance (weight) in recent months.


Weighted averages with different parameter a

Example of single customer with action DOWNGRADE . Grey curves are monthly curves

Weighted averages with different parameter a

Example of single customer with action CHURN . Grey curves are monthly curves

Figure 5: Weigted mean fuctions for two customers. Comparison of different parameter a value

For sections Outlier detection and removal and FANOVA only the case when $a = 1$ is used. The comparison of different a values is later investigated in the Classification part.

It is worth mentioning that although the dataset is not perfectly balanced class-wise, the difference in proportions is not very big at the same time. Therefore no measures are taken to tackle the slight class imbalance.

4 Analysis

This section is dedicated to the functional data analysis of the mobile data usage dataset.

4.1 Outlier detection and removal

Once functional data objects are created, it is highly important to perform an analysis of the outliers and make decisions on how these observations can be treated. This step has a big influence on further analysis and disregarding it can cause misleading results and conclusions.

For this particular dataset, the outlier detection procedure described in the subsection 2.4 is used.

The introduced outlier detection algorithm depends on several parameters.

- Type of depth measure;
- Number of bootstrap samples;
- The smoothing parameter for the bootstrap samples;
- Quantile to determine the cutoff from the bootstrap procedure;
- The α of the trimming (1% was used in the subsection 2.4).

There is no strict methodology constructed on how to select the optimal set of parameters for such an algorithm. Also, as there are no "true" or "false" outliers in the investigated dataset, the method's real performance in this given case can not be objectively measured. Thus the final set of parameters was obtained by experimenting with multiple different variations of it and performing the visual analysis.

After experimentation, such parameters were used for final outlier detection:

- Type of depth measure - Random Projections depth;
- Number of bootstrap samples - 100;
- The smoothing parameter for the bootstrap samples - 0.1;
- Quantile to determine the cutoff from the bootstrap procedure - 0.5;
- The α of the trimming - 0.1.

The outliers detected in each action group are displayed in the figure 6.

It can be noticed from the figure 6, that most of the outliers in all groups are related to unusually high mobile data usage. Nevertheless, several curves that represent a very low activity are also treated as outliers.

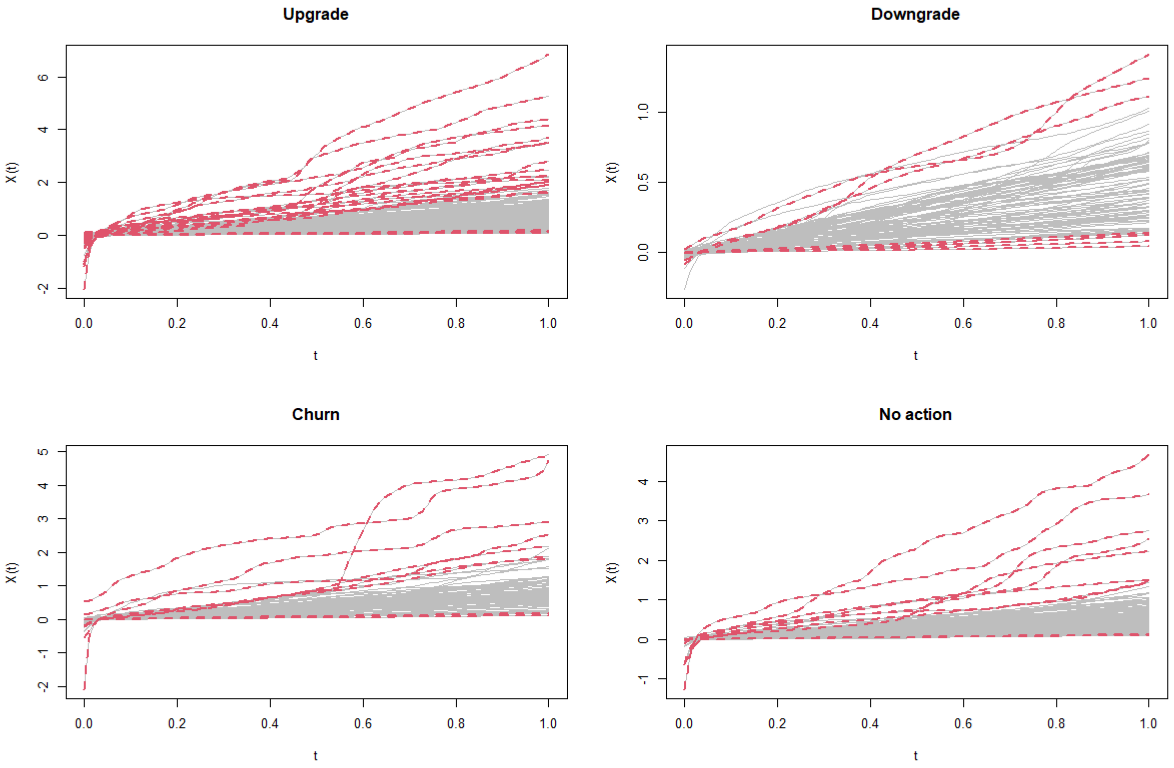


Figure 6: Outliers detected in each category

Also, it can be seen that in the group of "Downgrade" customers, the outliers are not as clear as in the other three groups. This comes as a consequence of this group being the smallest and the most homogenous one.

Regarding the other three groups, the detected outliers seem to be different from normal observations.

Once the outliers are detected, they are removed from the investigated dataset, and they do not participate in further analysis.

4.2 FANOVA

Before applying the classification algorithms, a necessary step is to check whether the groups of observations differ by some centrality measure. Although the thesis's aim is related to depth classifiers, there is not enough knowledge and methodology on analysis of variance which would be based on depth measures yet. Therefore FANOVA for functional means is used here to test whether the means of each group of observations differ statistically significantly.

When analyzing mean functions visually from the graphs 7 and 8, a separation between the mean functions of top groups (*Churn* and *Upgrade*) and the bottom groups (*Downgrade* and *No action*) seems to be large and probably significant. Whereas the functional means of *Churn* and *Upgrade* groups seems to be very close. Somewhat similar behaviour can be observed between the *Downgrade* and *No action* groups, although the mean curves increase the separation as t increases.

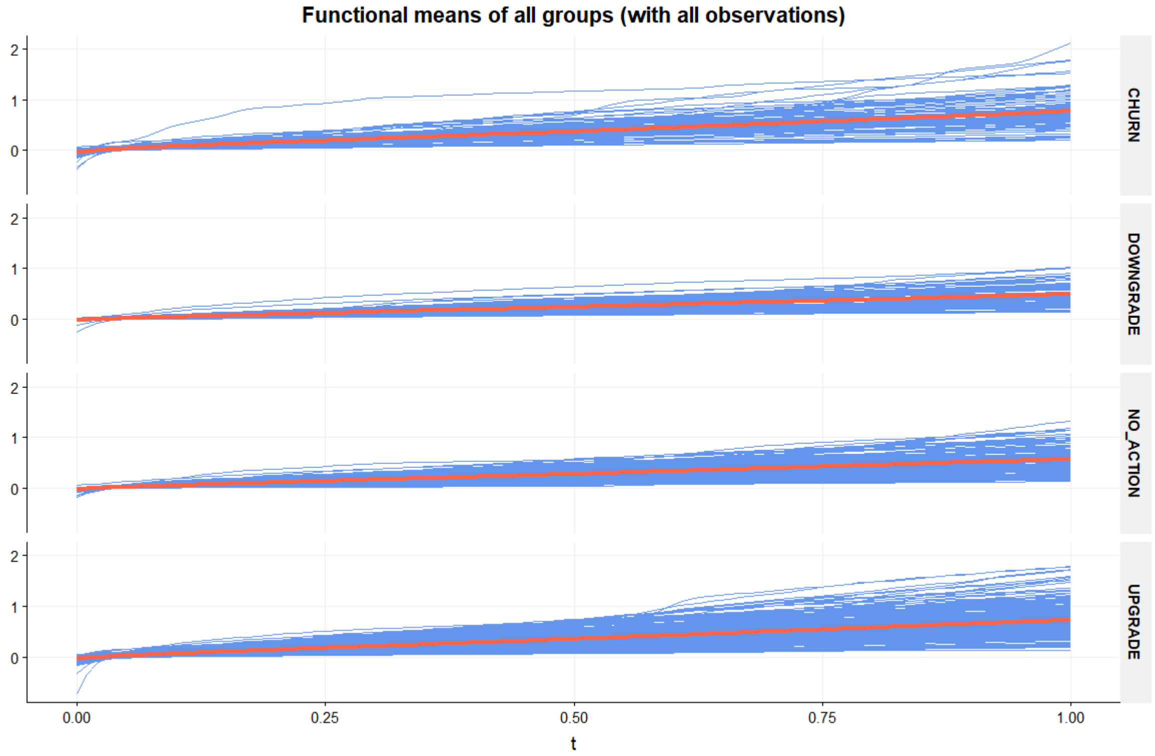


Figure 7: Functional means of all groups

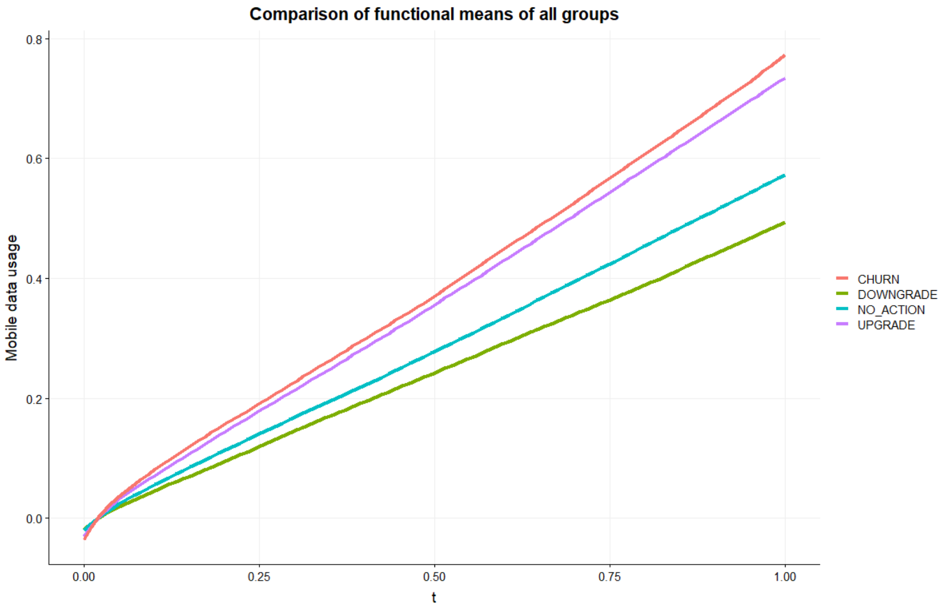


Figure 8: Comparison of functional means of all groups

Having these insights in mind, at first, the FANOVA tests for all four categories are applied. After that, the hypotheses are tested for each possible pair of groups.

So he first hypothesis is such

$$H_0 : \mu_{upgrade}(t) = \mu_{downgrade}(t) = \mu_{churn}(t) = \mu_{no\_action}(t), t \in [0, 1] \tag{13}$$

while the other six hypothesis are formulated around the pairwise functional mean equality.

Multiple tests are used from both FANOVA test groups introduced in the subsection 2.5. The table 4 contains the results of the used tests for several hypotheses tested. A more detailed explanation on the tests (including R implementation of them) is provided by Górecki and Smaga (2019).

Table 4: Results for 4 groups FANOVA (Description of the tests can be found in the table 8 in the Appendix)

Test	**All 4 groups**		**Upgrade vs. Churn**		**Downgrade vs. No action**	
	Test statistic	**p-value**	**Test statistic**	**p-value**	**Test statistic**	**p-value**
CH	1082.51	0	11.09	0.378	26.68	0.016
CS	1082.51	0	11.09	0.417	26.68	0.013
L2N	194.3	0	2.7	0.388	10.55	0.016
FN	22.29	0	0.78	0.389	5.55	0.017
Fb	22.29	0	0.78	0.408	5.55	0.015

As seen in the table 4, the hypothesis 13 with all four groups is strongly rejected, meaning that at least one pair of functional means of different groups is statistically significantly different. As this test does not bring much information, thus the pairwise tests are performed.

As it was expected based on the visual analysis, the functional means of *Churn* and *Upgrade* groups do not differ statistically significantly with the confidence level $\alpha = 0.05$. The equality of these two mean functions could imply that the customers' behaviour from *Churn* and *Upgrade* groups is fairly similar. This might mean that in most cases, the reason behind the churn and upgrade action is the same - data bucket being too small. Furthermore, if the customer's monthly data allowance is too small, then the action (either churn or upgrade) is just a consequence. Therefore it might be useful and wise to merge the two groups into a single one since the root cause of such actions is likely to be the same.

Also, it is worth mentioning, that the hypothesis $H_0 : \mu_{downgrade}(t) = \mu_{no\_action}(t)$ is rejected with the confidence level $\alpha = 0.05$. But if $\alpha = 0.01$ would be used, the H_0 could not be rejected. Thus it might be said that the difference between the mean functions of groups *Downgrade* and *No action* is statistically significant (with $\alpha = 0.05$), but it is not very big.

Regarding the rest of the four pairwise hypotheses, all of them were strongly rejected with p-values being very close to zero, meaning that in all four cases, the functional means differ statistically significantly.

The results of all pairwise hypotheses can be summarized using the matrix table 5.

Table 5: Summary of pairwise hypotheses testing results

Group	Upgrade	Downgrade	Churn	No action
Upgrade	-			
Downgrade	rejected	-		
Churn	not rejected	rejected	-	
No action	rejected	rejected (weakly)	rejected	-

Taking the results of the tests into account, an additional test for merged groups was performed. The null hypothesis for such test claims that the mean functions of two groups *Upgrade & Churn* and *Downgrade & No action* are equal. After performing the same tests, such hypotheses are rejected with p-value being almost equal to 0 for all tests, meaning the mean functions do not differ statistically significantly.

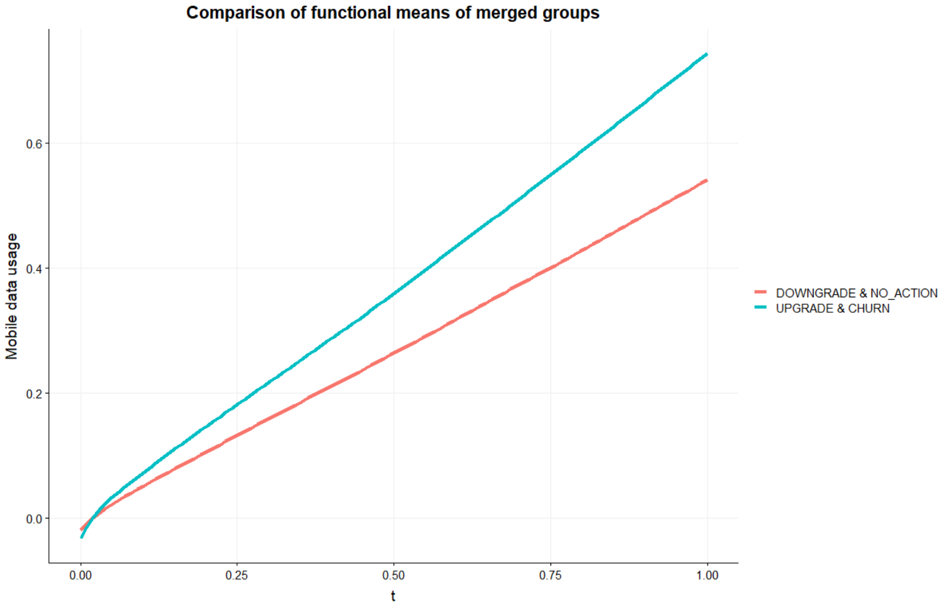


Figure 9: Comparison of functional means of merged groups

4.3 Classification

This section describes the functional classification analysis using the investigated dataset.

Based on the motivation given in the FANOVA section, the merged groups are used for classification, meaning that the dataset consists of observations with two labels: *Upgrade & Churn* and *Downgrade & No action*. As a result, a supervised binary classification problem is investigated.

For the sake of simplicity, the events are named as $A :=$ *Downgrade & No action* and $B :=$ *Upgrade & Churn*.

4.3.1 Classifiers used in the experiment

There are several dimensions using which the comparison is made in order to find the optimal classification method for classifying the customers based on their historical behaviour. These dimensions are such: classifiers, depth measures, weights of the weighted mean (aggregation function).

Classifiers that are based on DD-plots were introduced in the Functional Supervised Classification part. All mentioned approaches are applied to the upcoming customer action classification problem. The classifiers which are used are such:

- Maximum depth (MD)
- DD-classifier using polynomial function of degree 2 (DD2)
- DD-classifier using k nearest neighbour algorithm (KNN)
- DD-classifier using quadratic discriminant analysis (QDA)

Since all of the classifiers are based on functional depth measures, a comparison between different depths is made. So the mentioned classification methods will be based on these depths:

- h-modal (mode)
- Fraiman-Muniz (FM)
- Random projection depth (RP)
- Double random projection depth (RPD)

And the last dimension, which displays the importance of historical mobile data usage, is the aggregation function (or weights for mean function). The comparison between different weights combinations can bring significant knowledge on what kind of historical period is important when trying to predict an upcoming action. The investigated weights depend on the parameter a, for which five different values are tested and compared: $a \in \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$.

To sum up, 80 different variations of classification are used. The results of these all classifiers are summarized and compared in the later sections.

4.3.2 Classification of upcoming customer action

In this section, a detailed description is provided on how the classification analysis was performed.

At first, the cleaned dataset is randomly split into two parts: train and test set. In this analysis, 75% of the records from each group were treated as training observations, while the rest 25% was left for testing the performance of the classification algorithm. It is important to note, that same proportions of each group were retained in both train and test sets.

For each curve, depth with respect to *Downgrade & No action* and depth with respect to *Upgrade & Churn* are calculated. Then using these two depth metrics, every curve is represented on the DD-plot, where horizontal and vertical axis correspond to the depths of both groups, respectively.

The DD-plots (based on RPD depth) of train and test sets are displayed in the figure 10. There is no clear separation observed between the two groups in both DD-plots. Nevertheless, some zones seem to have a higher density of points from one group.

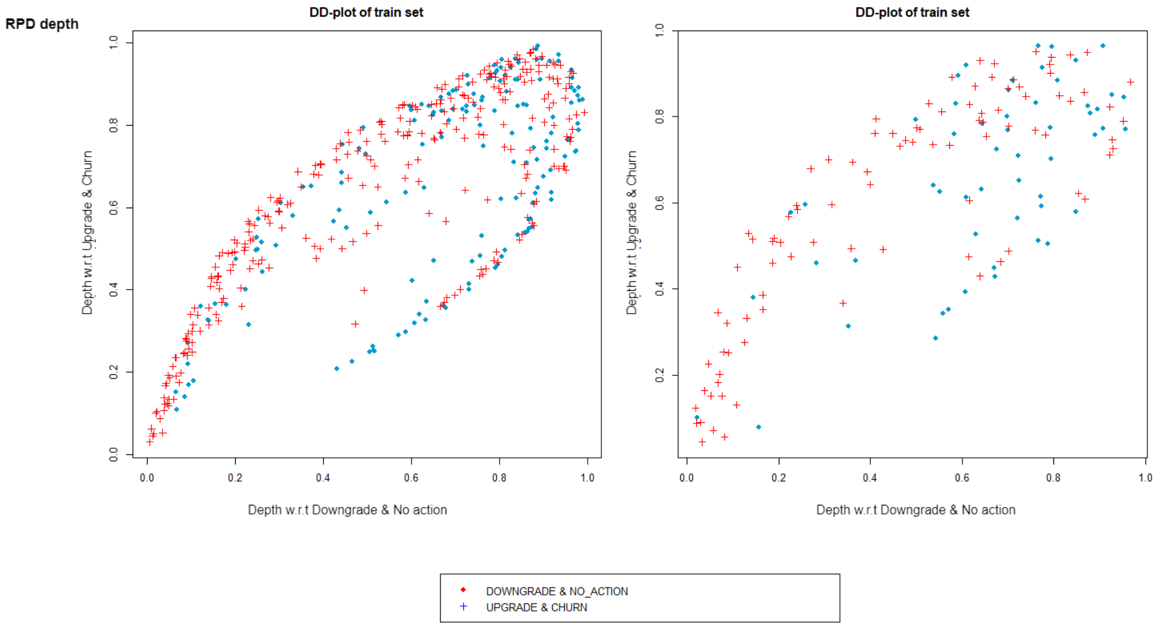


Figure 10: DD-plots of train and test sets (RPD depth).
DD-plots with all four depth methods can be found in the Appendix (15)

After splitting the dataset, training procedures are executed. The classifiers listed in the Classifiers used in the experiment are trained using the methodology that was introduced in the earlier stages of the thesis.

After training the algorithm, the classification rules are set. Let's consider the case, when RPD depth is used and $a = 0$ (see Figure 11).

The Maximum depth and DD2 classifiers, even though being simple, seems to do fairly well in this case. While QDA classifier behaves similarly to the mentioned ones, it is slightly

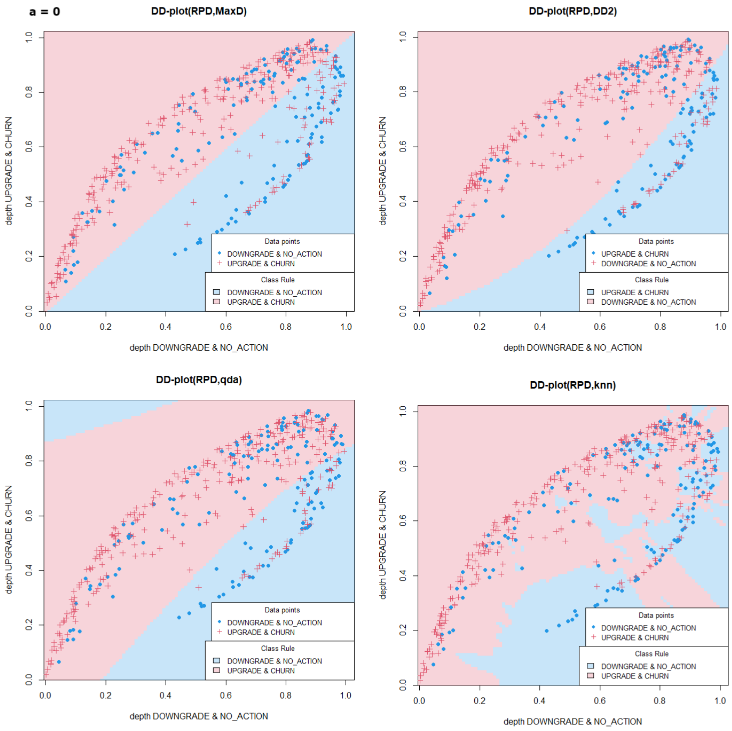


Figure 11: Trained classifiers. RPD depth, a = 0

biased towards the red group, which is *Upgrade & Churn* (Action B). QDA classifier tends to assign the points on the diagonal to the *Upgrade & Churn* (Action B) category. Meanwhile, the KNN method "tries" to be very precise, but in this case, it seems to be overfitting.

[Note: figures of other trained classifiers can be found in the Appendix A Figures.]

4.4 Classifier performance

The performance of trained algorithms is evaluated using the test set. Accuracy, sensitivity and specificity are the metrics, based on which model's efficiency is evaluated. In the case of this problem, the sensitivity reflects how good the classifier is at predicting the Action A (*Downgrade & No action*), while specificity shows the efficiency at predicting the Action B (*Upgrade & Churn*).

Consider the same case when classifier uses RPD depth and $a = 0$. After predicting the group labels for the test set with four classifiers, confusion matrices 6 are retrieved along with the accuracy, sensitivity and specificity metrics in table 7.

On the test set, Maximum depth classification method displays the highest accuracy (0.693) while DD2 classifier performs the worse out of compared techniques. However, based on specificity, the KNN approach shows a good result (0.905), meaning that it is good at classifying the customers who are truly willing to upgrade o churn.

It hards to tell, which classifier is the best. It depends on the target that the user is

Table 6: MD classifier confussion matrix. RPD depth, a = 0

MD

		Actual	
		Action A	**Action B**
Prediction	**Action A**	28	19
	Action B	27	76

DD2

		Actual	
		Action A	**Action B**
Prediction	**Action A**	18	15
	Action B	37	80

QDA

		Actual	
		Action A	**Action B**
Prediction	**Action A**	17	11
	Action B	37	84

KNN

		Actual	
		Action A	**Action B**
Prediction	**Action A**	16	9
	Action B	39	86

trying to achieve in every scenario. A KNN classifier discussed above brings a lot of Type II errors, and it does not provide the best accuracy. Nevertheless, in some situations, it might be an acceptable tradeoff for high specificity, if the user wants to minimize the Type I errors.

Table 7: Classifier's performance measures on test set. RPD depth, a = 0

	Classifier			
	MD	**DD2**	**QDA**	**KNN**
Accuracy	0.693	0.653	0.6733	0.68
Sensitivity	0.509	0.327	0.309	0.291
Specificity	0.8	0.842	0.884	0.905

4.4.1 Comparison of the classification results

During the experiment, 80 different variants of classification models have been evaluated. The results are discussed in this section.

There are three main dimensions for classifier comparison. They might be called hyperparameters in this case:

1. Parameter a, which defines the weights for the aggregation function (mean).

2. Functional depth calculation method.

3. Classification method.

Further on the hyperparameters' influence to classification result is discussed.

Starting from the **parameter a**. This parameter adjusts the weights of mean function that is used to aggregate the 12 monthly curves into a single representative curve. As described in section Aggregation using weighted mean, the bigger a value is, the more weight recent months receive compared to the older months. An objective of such hyperparameter is to test, whether the past months lose the informational value, that they bring to the classification task.

No significant relations between the parameter a and the accuracy increase or decrease are observed in the boxplot 12. However, it can be said that the median accuracy with parameter a being equal to 1 is slightly higher than the median accuracy of any other "competitor". Although this fact is not enough to confidently claim, that classification with $a = 1$ is performing better than others; it can safely be said, that the parameter a has a significant influence on the classifiers' performance.

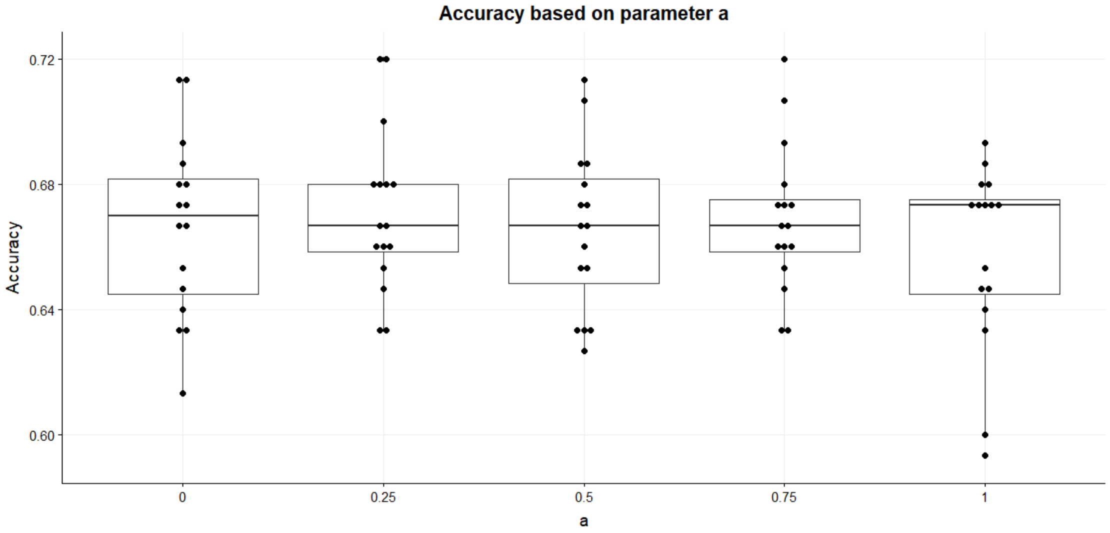


Figure 12: Accuracy boxplot

The second hyperparameter is **depth calculation method**. Four different depth options were investigated: Fraiman and Muniz (FM), h-modal, Random Projections (RP), Double Random Projections (RPD).

The accuracy results of classifiers with different depth notions are presented in figure 13.

In this case, there is a clear difference in accuracy distribution between the four depths. FM and RP showcase high variability with the amplitude of change being equal to 0.127 and 0.12, respectively. Also, the medians of these depths methods are lower than the h-modal and RPD depths' median accuracy. The distribution of h-modal and RPD depths accuracy

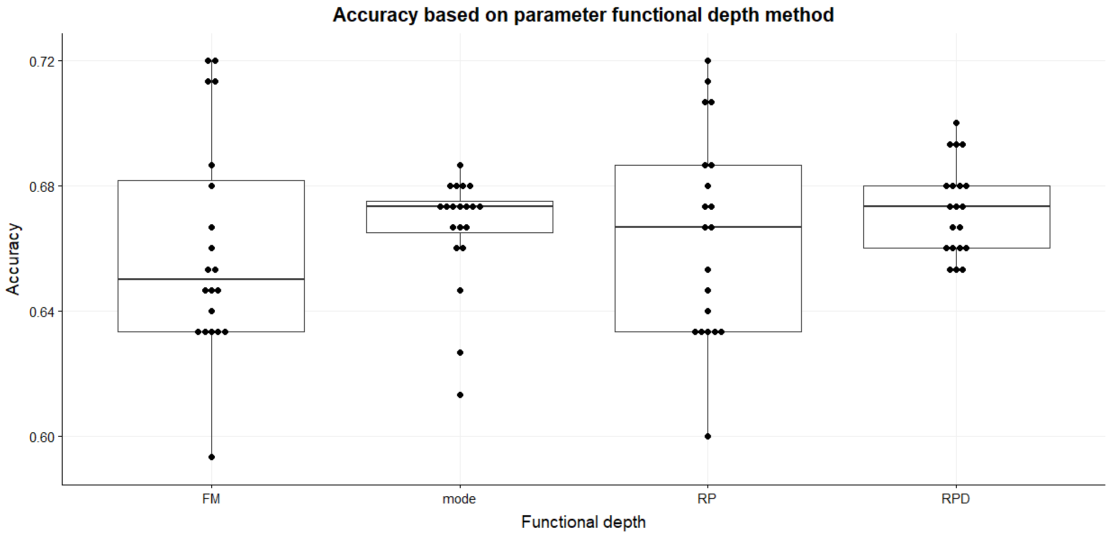


Figure 13: Accuracy boxplot

is much more squeezed, showing a greater consistency of classifiers using these two depth calculation methods. Therefore, in this, it could be claimed, that in most cases h-modal and RPD depths outperform the FM and RP depth-related classifiers when it comes to the accuracy. Also, a more general conclusion could be drawn, stating that the different functional depth calculation methods significantly impact the binary supervised classifier's performance.

The last investigated hyperparameter is the **classification method**. This one is interesting since it comes as the last step of defining the classification rule, therefore it is expected to be highly influential. The boxplot 14 contains the accuracy distributions with respect to different classification methods used to classify points in the two-dimensional DD-plots. The classifiers that were used are maximum depth (MD), the polynomial of degree 2 (DD2), quadratic discriminant analysis (QDA) and k-nearest neighbour (KNN).

As seen from the figure 14, according to the classification accuracy, the MD classifier, even though being the most trivial one, outperforms the rest of the pack. Concerning the accuracy, the second-best is the DD2 classifiers, which is also reasonably straightforward compared to the KNN and QDA methods. However, the variation of KNN classification accuracy is quite big, and few of the top classifiers even outperform the DD2 with respect to accuracy. Nevertheless, KNN likely tends to overfit as it might be reasonably sensitive, as seen in the previously discussed example 11.

Apart from that, it shows that in the real world problems like such, when the two groups share similar behaviour, the simple MD classifier is able to achieve the best accuracy among the compared methods.

Two models achieved the highest overall accuracy, which is 0.72. These models are Maximum Depth classifier with Fraiman and Muniz depth when $a = 0.25$ and $a = 0.75$. The

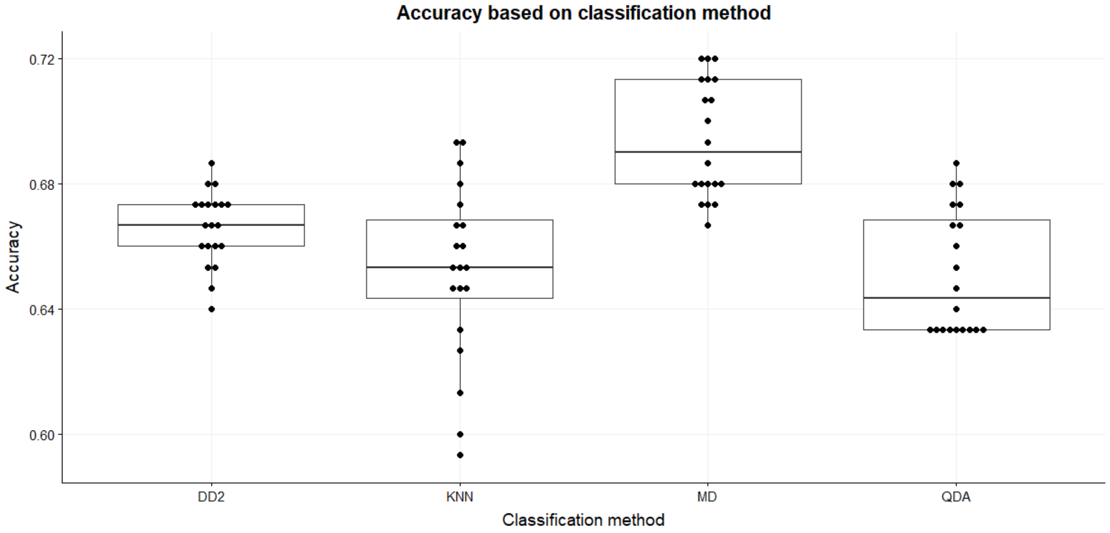


Figure 14: Accuracy boxplot

same models displayed the highers overall sensitivity of 0,71 while their specificity is equal to 0,726. The results of all models with respect to accuracy, sensitivity and specificity can be found in the B.

5 Conclusions

The thesis presents the Functional Data Analysis approach towards classifying the telecommunication company customers based on their behaviour when using mobile data.

Using FANOVA tests, it was demonstrated that the customers who are willing to upgrade or churn have no significant differences with respect to their average behaviour. This allows us to conclude that the churn action is often driven by the customer's need for an upgrade.

Eighty variations of depth-based classifiers were explored and compared to determine the model that produces the best result in upcoming customer action prediction. Three main parameters and their influence on the classification model result were investigated. The comparative analysis showcased that:

- The parameter a, which controls the weights in the aggregation function, significantly influences the classifiers' performance.
- h-modal and Double Random Projection depths produced more stable and consistent accuracy when used for depth-based classifiers.
- The Maximum depth classification technique resulted in the highest median and overall accuracy.

In the end, the most efficient were two Maximum depth models based on FM depth with a=0.25 and a=0.75. Both of them achieved the highest overall accuracy of 72%.

Possible improvement that could be addressed in the future is related to a different way of aggregating the historical curves to preserve customer behaviour dynamics throughout the year better. Also, multiple other factors might determine customer action. Therefore the current models might be improved and adjusted by adding other non-usage-related information into it.

References

Amin, A., Anwar, S., Adnan, A., Nawaz, M., Alawfi, K., Hussain, A., and Huang, K. (2017). Customer churn prediction in the telecommunication sector using a rough set approach. *Neurocomputing*, 237:242 – 254.

Aspirot, L., Bertin, K., and Perera, G. (2009). Asymptotic normality of the nadaraya–watson estimator for nonstationary functional data and applications to telecommunications. *Journal of Nonparametric Statistics*, 21:535–551.

Ben Slimen, Y., Allio, S., and Jacques, J. (2017). Anomaly prevision in radio access networks using functional data analysis. pages 1–6.

Cuesta-Albertos, A., J., Febrero-Bande, M., and Oviedo de la Fuente, M. (2016). The ddg-classifier in the functional setting. *TEST*, 26(1):119–142.

Cuevas, A., Febrero-Bande, M., and Fraiman, R. (2004). An anova test for functional data. *Computational Statistics Data Analysis*, 47:111–122.

Cuevas, A., Febrero-Bande, M., and Fraiman, R. (2007). Robust estimation and classification for functional data via projection-based depth notions. *Computational Statistics*, 22:481–496.

European Parliament, C. o. t. E. U. (2002). Universal service directive. `https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32002L0022`.

Febrero-Bande, M., Galeano, P., and Gonzãlez-Manteiga, W. (2008). Outlier detection in functional data by depth measures, with application to identify abnormal nox levels. *Environmetrics*, 19:331 – 345.

Fraiman, R. and Muniz, G. (2001). Trimmed means for functional data. *TEST: An Official Journal of the Spanish Society of Statistics and Operations Research*, 10:419–440.

Ghosh, A. K. and Chaudhuri, P. (2005). On maximum depth and related classifiers. *Scandinavian Journal of Statistics*, 32(2):327–350.

Górecki, T. and Smaga, u. (2015). A comparison of tests for the one-way anova problem for functional data. *Computational Statistics*, 30.

Górecki, T. and Smaga, u. (2019). fdanova: an r software package for analysis of variance for univariate and multivariate functional data. *Computational Statistics*, 34.

Huang, B., Kechadi, M. T., and Buckley, B. (2012). Customer churn prediction in telecommunications. *Expert Systems with Applications*, 39(1):1414 – 1425.

J.A., C.-A., M., F.-B., and de la Fuente M., O. (2017). The ddg-classifier in the functional setting. *TEST 26*, page 119–142.

Li, J., Cuesta-Albertos, J. A., and Liu, R. Y. (2012). Dd-classifier: Nonparametric classification procedure based on dd-plot. *Journal of the American Statistical Association*, 107(498):737–753.

Li, J. and Liu, R. Y. (2004). New nonparametric tests of multivariate locations and scales using data depth. *Statistical Science*, 19(4):686–696.

Liu, R. Y. (1990). On a notion of data depth based on random simplices. *The Annals of Statistics*, 18(1):405–414.

Liu, R. Y., Parelius, J. M., and Singh, K. (1999). Multivariate analysis by data depth: descriptive statistics, graphics and inference. *Ann. Statist.*, 27(3):783–858.

Ramsay, J., Hooker, G., and Graves, S. (2009). *Functional data analysis with R and MATLAB*. Springer, New York, NY.

Ramsay, J. and Silverman, B. (2008). *Applied Functional Data Analysis*, volume 24. Springer, New York, NY.

Ramsay, J. O. and Silverman, B. W. (2010). *Functional data analysis*. Springer Science Business Media.

Wang, J.-L., Chiou, J.-M., and Mueller, H.-G. (2015). Review of functional data analysis.

Appendices

A Figures

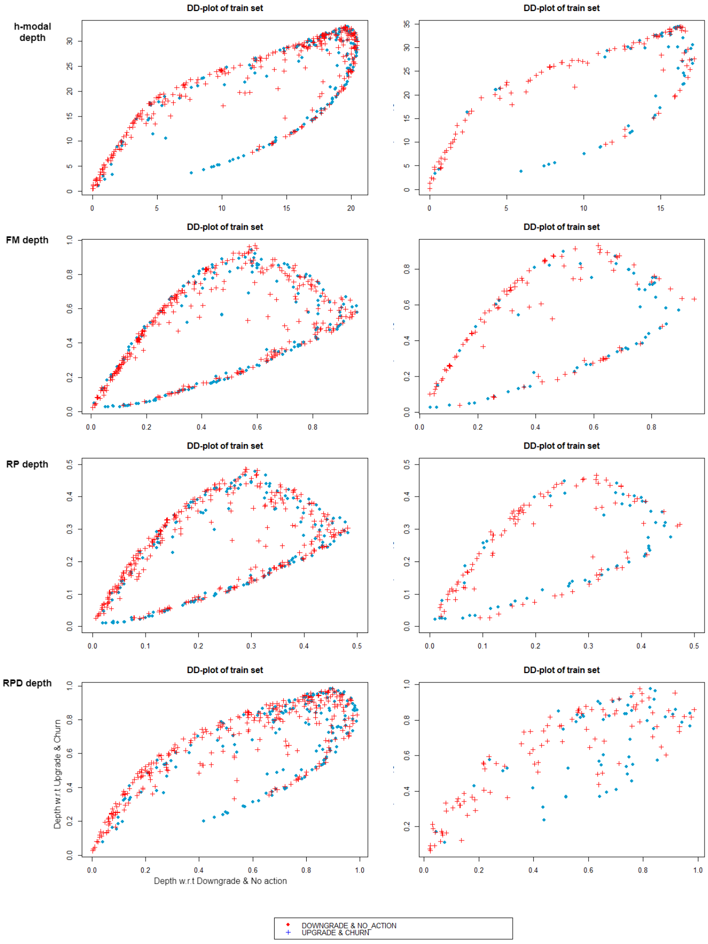


Figure 15: DD-plots of train and test sets (h-modal, FM, RP, RPD depths)

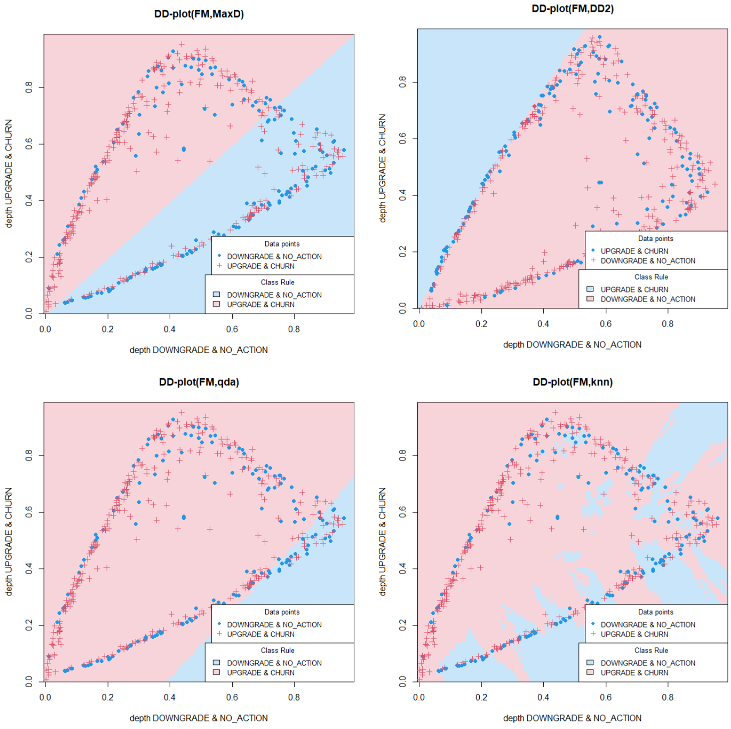


Figure 16: Trained classifiers using FM depth

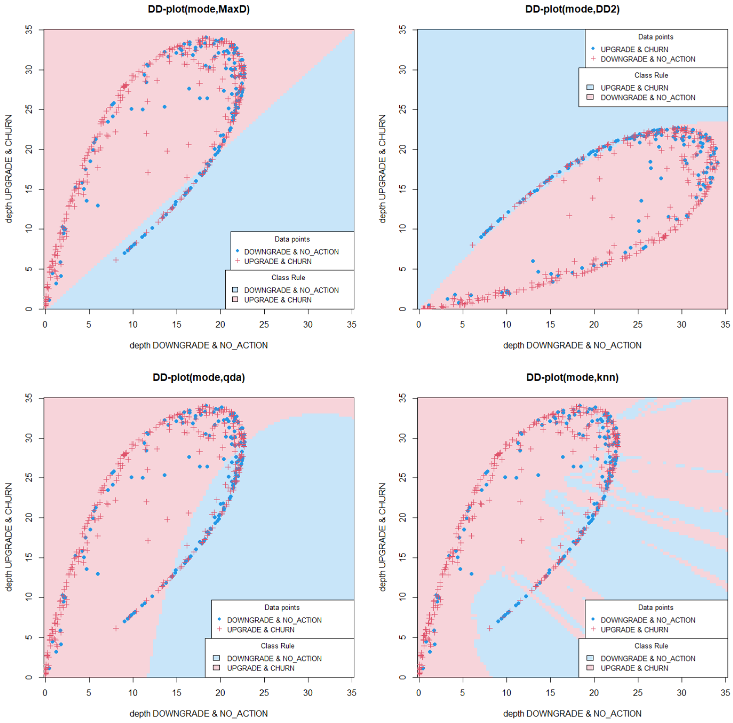


Figure 17: Trained classifiers using h-modal depth

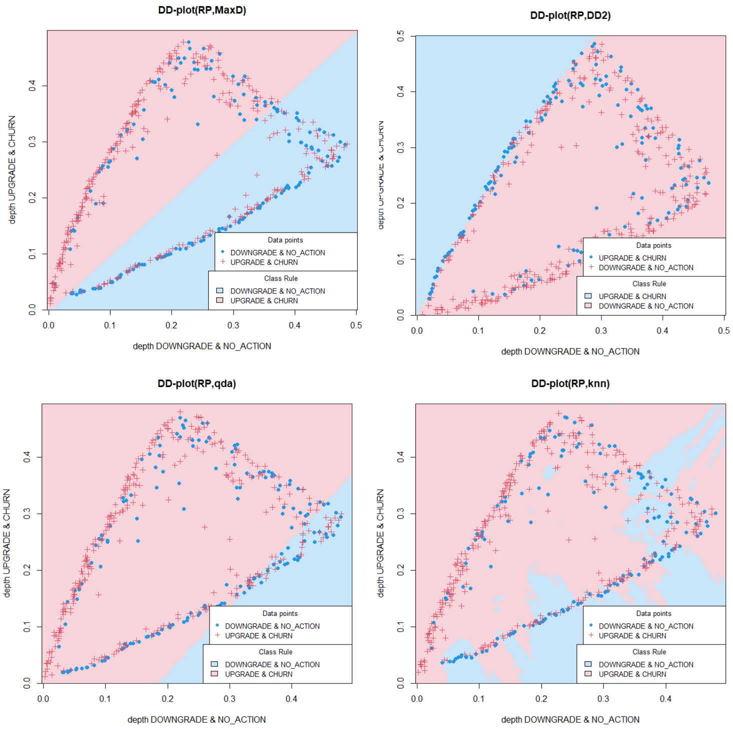


Figure 18: Trained classifiers using RP depth

B Tables

Table 8: Description of FANOVA tests

Test (short)	**Description**
CH	L^2-norm-based bootstrap test for homoscedastic samples
CS	L^2-norm-based bootstrap test for heteroscedastic samples
L2N	L^2-norm-based test with naive method of estimation
FN	F-type test with naive method of estimation
Fb	F-type bootstrap test

Table 9: Performance metrics of MD classifiers evaluated on the test set

a	**Classifier**	**Depth**	**Accuracy**	**Sensitivity**	**Specificity**
0,25	MD	FM	0,720	0,709	0,726
0,75	MD	FM	0,720	0,709	0,726
0	MD	FM	0,713	0,709	0,716
0,5	MD	FM	0,713	0,709	0,716
1	MD	FM	0,680	0,709	0,663
0,25	MD	RP	0,720	0,691	0,737
0	MD	RP	0,713	0,691	0,726
0,5	MD	RP	0,707	0,691	0,716
1	MD	RP	0,687	0,691	0,684
0,75	MD	RP	0,707	0,673	0,726
0	MD	RPD	0,693	0,509	0,800
0,25	MD	RPD	0,700	0,491	0,821
0,5	MD	RPD	0,680	0,418	0,832
0,75	MD	RPD	0,680	0,400	0,842
1	MD	RPD	0,680	0,400	0,842
0,25	MD	mode	0,680	0,291	0,905
0,5	MD	mode	0,673	0,273	0,905
1	MD	mode	0,673	0,273	0,905
0,75	MD	mode	0,673	0,255	0,916
0	MD	mode	0,667	0,236	0,916

Table 10: Performance metrics of DD2 classifiers evaluated on the test set

a	**Classifier**	**Depth**	**Accuracy**	**Sensitivity**	**Specificity**
0,5	DD2	RP	0,687	0,345	0,884
0,25	DD2	mode	0,680	0,291	0,905
0,25	DD2	RP	0,680	0,291	0,905
0,75	DD2	mode	0,673	0,327	0,874
1	DD2	mode	0,673	0,327	0,874
1	DD2	RPD	0,673	0,309	0,884
0,75	DD2	RP	0,673	0,291	0,895
1	DD2	RP	0,673	0,291	0,895
0	DD2	mode	0,673	0,255	0,916
0,5	DD2	FM	0,667	0,345	0,853
0,5	DD2	mode	0,667	0,291	0,884
0	DD2	RP	0,667	0,273	0,895
0,25	DD2	RPD	0,660	0,309	0,863
0,5	DD2	RPD	0,660	0,291	0,874
0,75	DD2	RPD	0,660	0,273	0,884
0,25	DD2	FM	0,660	0,182	0,937
0	DD2	RPD	0,653	0,327	0,842
0,75	DD2	FM	0,653	0,236	0,895
1	DD2	FM	0,647	0,255	0,874
0	DD2	FM	0,640	0,145	0,926

Table 11: Performance metrics of QDA classifiers evaluated on the test set

a	**Classifier**	**Depth**	**Accuracy**	**Sensitivity**	**Specificity**
0,5	QDA	mode	0,687	0,364	0,874
0	QDA	mode	0,680	0,327	0,884
0,25	QDA	mode	0,680	0,327	0,884
0,5	QDA	RPD	0,673	0,327	0,874
0	QDA	RPD	0,673	0,309	0,884
0,25	QDA	RPD	0,667	0,309	0,874
0,75	QDA	RPD	0,667	0,309	0,874
0,75	QDA	mode	0,660	0,309	0,863
1	QDA	RPD	0,653	0,273	0,874
1	QDA	mode	0,647	0,273	0,863
1	QDA	RP	0,640	0,055	0,979
0	QDA	FM	0,633	0,000	1,000
0	QDA	RP	0,633	0,000	1,000
0,25	QDA	FM	0,633	0,000	1,000
0,25	QDA	RP	0,633	0,000	1,000
0,5	QDA	FM	0,633	0,000	1,000
0,5	QDA	RP	0,633	0,000	1,000
0,75	QDA	FM	0,633	0,000	1,000
0,75	QDA	RP	0,633	0,000	1,000
1	QDA	FM	0,633	0,000	1,000

Table 12: Performance metrics of QDA classifiers evaluated on the test set

a	**Classifier**	**Depth**	**Accuracy**	**Sensitivity**	**Specificity**
0,75	KNN	RPD	0,693	0,364	0,884
1	KNN	RPD	0,693	0,327	0,905
0	KNN	FM	0,687	0,309	0,905
0	KNN	RPD	0,680	0,291	0,905
1	KNN	mode	0,673	0,400	0,832
0,25	KNN	mode	0,667	0,382	0,832
0,75	KNN	RP	0,667	0,382	0,832
0,75	KNN	mode	0,660	0,382	0,821
0,25	KNN	RPD	0,660	0,309	0,863
0,5	KNN	RPD	0,653	0,345	0,832
0,5	KNN	FM	0,653	0,309	0,853
0,25	KNN	RP	0,653	0,273	0,874
0,75	KNN	FM	0,647	0,309	0,842
0	KNN	RP	0,647	0,291	0,853
0,25	KNN	FM	0,647	0,218	0,895
0,5	KNN	RP	0,633	0,327	0,811
0,5	KNN	mode	0,627	0,400	0,758
0	KNN	mode	0,613	0,255	0,821
1	KNN	RP	0,600	0,273	0,789
1	KNN	FM	0,593	0,218	0,811