Master Thesis Business Analytics

Churn analytics in Utrecht Libraries

A data-driven approach using machine learning

Jeroen Hörters

Supervisors:

Ynformed

Ramstraat 27a
3581 HD Utrecht

Anton Kuijer

Ruben Peeters

Vrije Universiteit Amsterdam
Faculty of Science
De Boelelaan 1081a
1081 HV Amsterdam

Rikkert Hindriks

Ger Koole (Second Reader)

BiSC Utrecht

Fruitweg 48
3981 PA Bunnik

Ronald Huizer

Stefan de Bruin

Executive Summary

This research attempts to predict customer churn based on three different machine learning models: Decision trees, Random Forests and Extreme Gradient Boosting. To make this prediction, datasets from the libraries on customers, collection and loan activity were used, alongside publically available neighbourhood data.

Out of the three models, the Gradient Boosting model performs best, achieving a recall of 91.4%, while still retaining an accuracy of 80.3%. As the positives are very much a minority class, these results are very promising. Looking at the variables that are relevant to this model, however, the most important ones coincide with the currently known risk factors for churn within libraries. Therefore, the model can not immediately be used to change existing library policies regarding membership retention.

Contents

1. Introduction

The public libraries in Utrecht have experienced a decline in membership numbers for a number of years. Fewer people read books and of those people even fewer use physical media to do so. However, libraries offer more services than just loaning books. Examples of other services are: Loaning e-books, offering a place to read newspapers, facilitating study spaces, organizing activities on all kinds of subjects and so-called "Human Libraries" where people with some interesting trait, job or hobby offer some of their time to answer all kinds of questions to an audience. All these secondary services contribute to the libraries' main purpose: creating social value.

Even though the library is focused on social value, being a financially healthy system is still of major importance. Contrary to popular belief, the income through members is relatively low compared to the subsidies that the libraries receive each year; generally around 15% of total income comes through users, consisting of membership and late fees. The remaining funds come from subsidies, which are largely based on the social value that the libraries create for the people in their proximity.

BiSC, the Library Service Centre, is the company that manages all data generated by the Utrecht libraries. They also manage many other activities related to the libraries, such as distribution of the books between the libraries and developing various innovation plans. Part of their digital innovation roadmap is to start using the large amount of data BiSC has stored on the libraries' members to create social impact for the libraries in a data-driven way.

Initially, there were many different ideas for ways to accomplish this innovation. For example, using loan data and the collection database to create a recommendation system for the libraries that is superior to the ones that are currently in use or devising a way to detect low-literate people in order for the libraries to find new participants for their in-house literacy courses.

After investigation of these possibilities and extensive discussions with BiSCs employees, it was determined that a different option had a better perceived chance at succeeding, namely using data-driven methods to analyse the churn patterns within the Utrecht libraries. The customer churn rate, also known as the attrition rate, is an often-used term with varying definitions. However, the essence of the definition is often the same. (Galetti, 2015) has an elegant and general definition: "… attrition rate is a calculation of the number of individuals or items that vacate or move out of a larger, collective group over a specified time frame.". This is general in

the sense that churn does not necessarily refers to customers, it may also apply to other groups of people, for example employees. The churn rate can be calculated by dividing the customers that have moved out in a given time period by the total number of customers that were present at the start of this time period. The following research question was constructed:

"How well can we predict the probability of a customer ending their membership in the next six months?"

Answering this question gives insight in the factors that contribute to the churn within libraries and will help the libraries in targeting specific groups of members with relevant secondary activities.

In Chapter 2, we look at the data used in this research. Firstly, we will take a closer look at the different datasets that were available for this research and describe their structure, features and size. Next, we explore the data to get more familiar with their intricacies. Subsequently, in Chapter 2.3, we discuss the steps that were taken to prepare the data for the machine learning part of the research and in Chapter 2.4 we discuss the experimental setup and the final dataset that was derived to use with the machine learning models.

In Chapter 3, we first discuss these machine learning models to see how they were first created and how they work. Next, we take a look at some performance measures with which we can compare the different models. In Chapter 3.3, we look at some benchmark predictions that we can use as a comparison as well. Lastly, we will discuss the methods of parameter optimization used in Chapter 3.4.

In Chapter 4, the results of the various machine learning models will be compared to both each other and to the various benchmarks that were constructed earlier after which the research question can be answered. Finally, the conclusions of this research will be discussed in Chapter 5, after which some recommendations will be stated (Ch. 6).

2. Data

In this chapter, we will look at the various datasets. Firstly, we will discuss their origin, their structure and their importance to answer the research question. Secondly, some initial data analytics techniques were applied to answer some business intelligence related questions about the data in order to get familiarized with it. Next, we will look at the different steps that were taken to improve the data quality by cleaning it and removing unneeded columns. Finally, the experimental setup that was designed will be described, as well as how the data needed to be manipulated in order for it to fit into this setup.

2.1. Datasets

In order to conduct this research, BiSC was able to provide several data files from the library data system, which fall into three categories. These categories are customer data, loan data and collection data. Additionally, some data on different neighbourhoods in the Utrecht province was retrieved from the public online database of the Dutch National Statistics Centre[1]. In this section, these datasets will be described in more detail. A full overview of all columns for these datasets can be found in the appendix.

2.1.1. Loan Data

The dataset concerning the book loans was provided for four years, from 2013 through 2016, in separate files per year. The number and ordering of the columns is the same for all files, so they can all be processed in the same way. This dataset describes for each loan, reservation and re-loan, the location, item and member that was involved. The only aspect that is not described by the data in detail is time. Only the date that the item was loaned is specified, not the exact time. Not knowing the loan time is not expected to be essential, but these times might have contained useful information for the machine learning models. The total number of loans across these four years add up to over 22 million entries across 10 columns.

2.1.2. Customer Data

The customer database spans all customers that were ever entered into a digital system across all libraries that are managed by BiSC. These members are called “actors”. Actors span not only people that have memberships, but also organisations, such as companies and schools. The

[1] Centraal Bureau voor de Statistiek (CBS)

actor ID is the connecting identifier between the customer and loan data. The customer data is very specific, containing the gender, date of birth and the ZIP code for each member.

Regarding the membership of the customer, we can see when the member registered their membership and which membership they have. The data also shows the membership costs and the duration of the membership.

Finally, there are also some less useful columns, regarding the customer "role". This could mean that the customer is part of a school, or perhaps a library employee. Since these members are exactly not those we are interested in analysing, these columns don't seem very useful. In Chapter 2.3 we see that these columns will not be used in the machine learning part of this research. The dataset contains about 400 thousand entries across 18 columns.

2.1.3. Collection Data

The collection database was also provided as a CSV-file and contains information on all items present in the various collections of all libraries managed by BiSC. It contains nearly 1.8 million items. There are many available columns, eighty in total; however, not all of those are of sufficient usefulness to the research or of sufficient quality. Moreover, many columns contain no data, or the values are the same for each item in the collection, rendering them useless. Also, the meaning of some columns is very vague or even unknown to the BiSC employees. This can be attributed to the current database being a result of older databases of different locations being merged after these locations joined BiSC.

2.1.4. Neighbourhood Data

The neighbourhood dataset consists of 124 columns that describe different aspects of each neighbourhood. These aspects contain demographic statistics, average distance to various facilities and many more. This dataset is freely available for anyone to download through the website of the Dutch bureau of statistics, the CBS (Centraal Bureau voor de Statistiek).

To be more precise, the neighbourhood dataset consists of population fractions based on age (14-, 15-24, 25-44, 45-64, 65+), marital status and heritage (Dutch, Moroccan, Surinam etc.), living standards, such as number of cars and average income, and demographic statistics, such as births and deaths. Also, there are a lot of columns regarding the average distance to the nearest facilities, such as hospitals, supermarkets, cinemas, and, fortunately for this research, libraries. A complete list of all the available columns can be found in the appendix.

Unfortunately, not all features are available for all neighbourhoods. This can be because they have not been collected, they have not been added into the database, or they might even be confidential.

2.2. Data Exploration

After receiving the datasets as described in the previous section, the initial goal was to gain familiarity with the data by means of exploration, in order to be able to create better features for the machine learning portion of this research. BiSC had already formulated a list of questions about the data, which was a good starting point to gain this familiarity. In this section, the results of the data exploration will be discussed by answering the following questions:

- How long do memberships last on average?

As seen in Figure 1, most memberships are terminated within the first year. Furthermore, most other memberships are ended in the first few years, but we can see that there is no critical value for membership cancellation. As such, we see that memberships are still ended after many years, even though this happens less frequently than with the shorter durations.

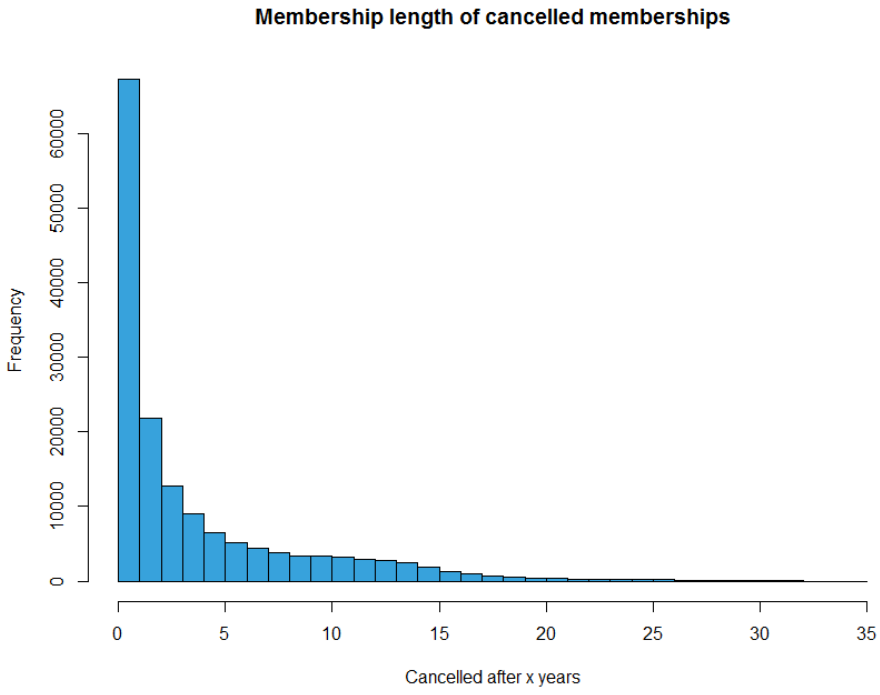

Figure 1: Bar plot of the membership lengths of memberships that were ended.

- What is the loaning behaviour of members that end their membership?

In Figure 2, every past (left) and current (right) member is plotted with their membership length and their average number of loans per year, with trend lines for each of the two plots. In the left plot, we see that the behaviour of people that ended their membership early does not seemingly differ from people that have been a member for multiple years and end their membership, because of the nearly horizontal trend line. In both graphs in Figure 2 we see that the data is dense around zero for all membership durations, since the trend lines are near zero for both. For the active membership, this average number of loans does increase above zero in the first years, but seems steady after that. Therefore we can conclude that people that are still members today do loan more than people that ended their membership used to during their membership.

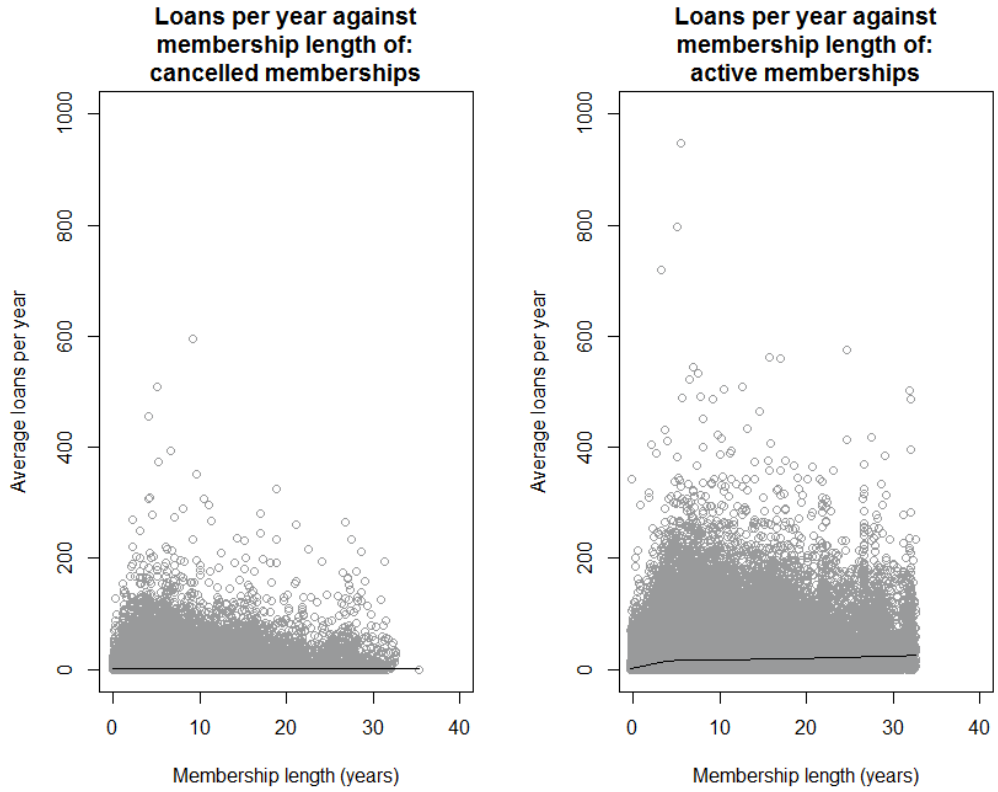

Figure 2: Scatter plots of average loans per year of previous and current members against membership duration

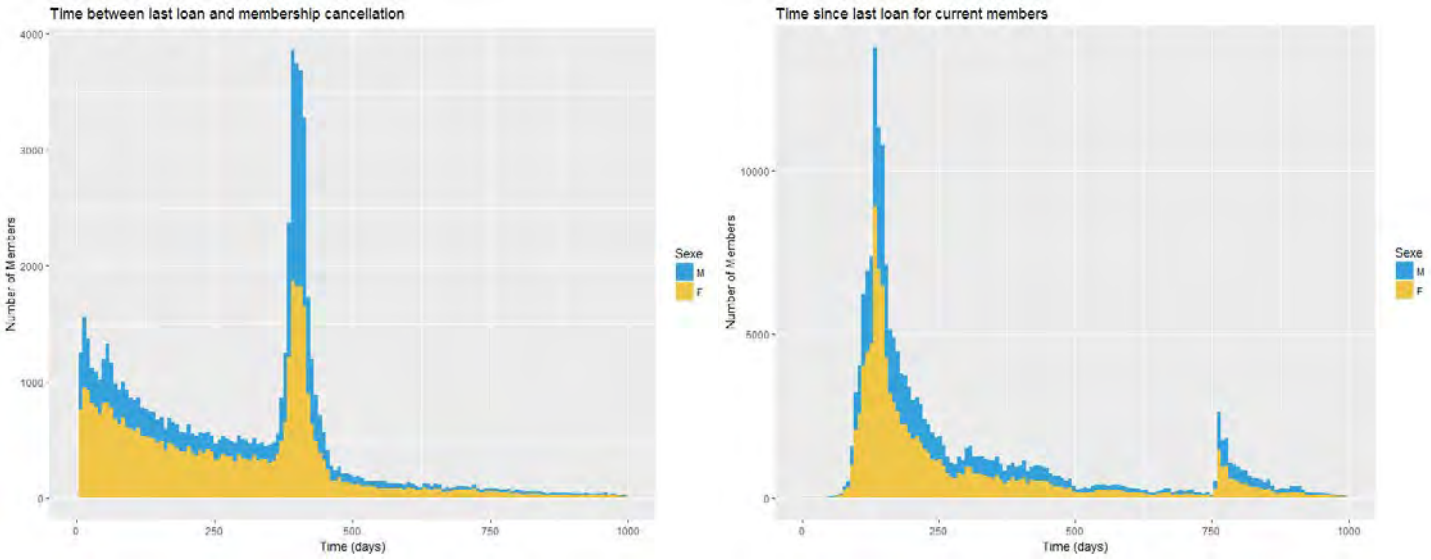

Figure 3: Time since last loan for both cancelled (left) memberships and current memberships. For the cancelled memberships, the time was calculated from the last loan until cancellation. For the current members, the cutoff date was set to the 12th of May 2017.

- How long have people that ended their membership not loaned for before they ended their membership?

In general, the distribution of time since last loan looks similar between current members and past members as can be seen in Figure 3. However, there is an additional peak between 300-400 days in the group that ended their membership. This could be explained by people realizing that they are still a library member after getting an invoice for their membership after a year of inactivity, leading them to end their membership. For the current members, there is an interesting peak around 750 days. This is probably due to some promotion that led to new members who afterwards never loaned again, but the true cause of this peak is unknown, even to the BiSC employees. So these members still have a membership, even after not loaning a single item in the last two years.

- How many people started a new membership after cancelling a membership in the past?

Unfortunately, the most detailed personal data that was available for this research was postal code data. A single code corresponds to multiple households and within a household there can be multiple memberships. Therefore, there is no way to answer this question with the available data.

- At what age do children become a member? Does this differ between boys and girls?

Most children start their membership before the age of seven. This holds for both boys and girls. A difference in behaviour between genders starts around the teenage years, during which more

girls have a membership than boys. These findings are shown in Figure 4. When we look at adults, this difference between genders becomes even more pronounced.

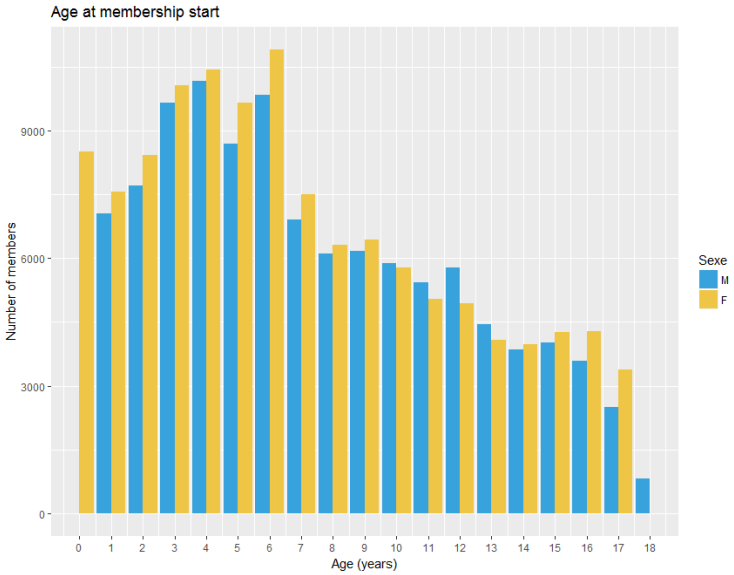

Figure 4: Age at which youth members start their membership. Parents are allowed to start library memberships for their children from birth, explaining the many babies and toddlers with memberships

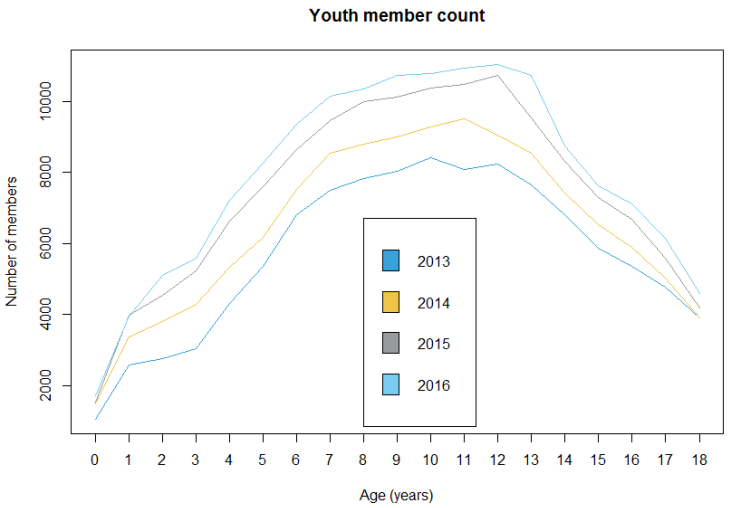

Figure 5: The number of youth members over the years. We can see that the number of members has increased every year for this age group.

- How have the youth membership numbers changed over the years?

The data shows that the number of younger member has increased significantly over the last years, as can be seen in Figure 5. This increase can be observed for both boys and girls. However, from ages 14 and up, as mentioned previously, we see that more girls have a membership than boys, and the number of members decreases as the age of the members increases.

- How much do youth members loan?

Young members experience their peak in loan activity somewhere between the ages of seven and eleven. Past those ages, the number of loans per year decreases drastically (Figure 6). As with the number of members, this is also the age during which girls start loaning more books than boys do. Over the years, however, the total number of loans each year stays about the same. Therefore, the average

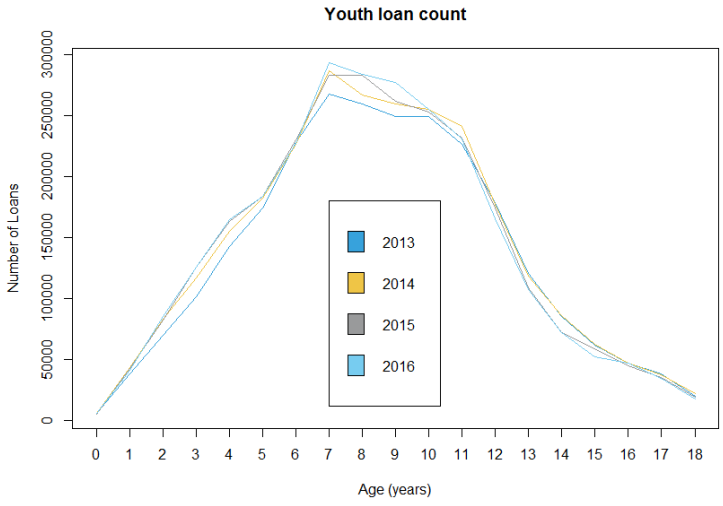

Figure 6: The total number of loans for members of a certain age for each of the years. We can see that these numbers have barely changed.:

number of loans per member actually decreases, since the number of members increases each year, as we saw previously.

- How many members are inactive?

From the approximately 400 thousand members, more than 25% has not loaned a single item since the start of 2013. About 88 thousand of these 115 thousand members have already ended their membership at the end of the data recording. After removing the members that started their membership after the start of 2013, we end up with 22 thousand members that have been a member throughout this four year period, but haven't loaned a single item.

As a final remark, we can take some findings from publically available data on the performance of libraries. From this national data provided by the CBS we see that only 15% of the library income comes from the customers directly. All the other funding comes from the state or the province. It is already hard to completely fund the libraries, so any innovation should either result in additional revenue, or a more efficient use of the current funds.

2.3. Data preparation

In this section, we will look at all different pre-processing steps that were taken for each of the datasets in order for them to be ready for the machine learning step of the research.

2.3.1. Loan data

Besides combining this data into a single file, some further preparation was required. Some of the columns were immediately removed from the data, since they would not be of any use, as mentioned in Section 2.1.2. Furthermore, several other columns could be removed after using them to filter out some rows that are of no use to this research. The usefulness of columns was thoroughly discussed with the data experts at BiSC. Some examples:

- "aktie" column: This column tells us whether an entry is a loan, a reservation or a renewal of a loan. For this research, we are only interested in loans, so all rows with different values could be removed, after which the column was deleted.
- "materiaal" column: This column describes the type of media that was loaned. There are some descriptions that are currently invalid and should not occur in the data. After filtering the entries, this column could also be removed.

In total, the following columns were removed after the previous filtering steps:

- “instantie_id” - ID
- “punt” – ID of the loaning terminal within a location.
- “bron” – Says nothing
- “aktie”
- “materiaal”
- “scatnr” – Item category
- “titelnr” – ID of the loaned item

Using these filtering techniques, the result was a single file containing all the loans in the years 2013-2016. It contains approximately 22 million entries with the following columns:

- Date
- Customer ID
- Location

After the actor data was fully processed as well, a large number of entries could be removed that belonged to actors that would not be considered, due to them having cancelled their membership before 2013. This action brought the total number of entries down to 19 million.

2.3.2. Actor Data

The actor data, or member data, contains information on all current and past library members from all libraries in the Utrecht province. As with the loan data, some cleaning and filtering needed to be done before the data was ready to be used in machine learning.

First off, there were a number of columns with statistics on loans in 2016 and 2017 which were deleted, since data on 2017 would not be used, and statistics on 2016 would be developed later along with the other years.

Next, more actors could be removed based on the “Sexe” and “Lid.status” columns. For the sex column, there were initially 4 values: M, V, I and 0. The actors with values I and 0 were removed, because these are memberships that concern schools, employees, businesses and other sources that are not of interest for this research. The “Lid.status” indicates the membership status of

each actor. There are some values of this column that indicate an invalid entry. These entries were also removed from the data.

Finally, there were some columns that concern the "role" of an actor. After consulting the data engineers, it was found that only one role would be valid for this research. All other columns concerning roles were deemed useless and were removed. After this initial step the following columns were left:

- "Actor.id" – ID
- "Vestiging" – Location where the membership was activated
- "Geboortedatum" – Date of Birth
- "Sexe" – Sex
- "PC.6" – ZIP Code
- "Abonnement.Omschrijving" – Membership type
- "Inschrijfdatum" – Date of membership registration
- "Uitschrijfdatum" – Date of membership termination
- "Ab.prijs" – Membership price per year in cents
- "Ab.periode" – Membership duration in months

After this initial preparation, more steps were needed. Firstly, the number of different membership types needed to be reduced. Initially, there were over 120 different types of memberships, and using all of those in machine learning would probably not work very well. Most of these membership types only have a very small number of members that belong to them. Due to their small size, the individual membership types to not have enough power to contribute to this model. Also when using dummy variables, this would result in very sparse features, which have the same problems (Hastie, Tibshirani, & Friedman, 2009). Therefore, some dimensionality reduction technique needed to be used. Memberships without any active members were removed and some small membership types were changed to other, larger types that are equivalent in price and permissions, such as number of books that can be loaned.

Next, the ZIP codes needed to be simplified. The ZIP codes in the data are so-called PC.6 codes. In the Netherlands, this is the most detailed version of the ZIP code.. A single PC.6 code generally

covers a single street and is of the form *"1234 AB"*. However, the neighbourhood is linked to PC.4 codes and is of the form *"1234"*. It is very simple to convert PC.6 to PC.4 by removing the letters from the PC.6 code. During this conversion process, it became apparent that there were instances of actors with invalid PC.6 codes. These actors were removed. Closer inspection showed that there were also actors with invalid PC.4 codes. These were either non-existing, because they are not in use, or codes of PO boxes, which have no physical location, so no neighbourhood data can be linked to them. These actors were also removed from the data as it only concerned a very small part of the total data. Now, with only valid PC.4 codes left, the actors table and neighbourhood data table can be easily merged in the future.

After close inspection of the registration, termination and birth dates of the actors, some anomalies were detected that needed to be filtered out. Actors without birth or registration dates were removed, as were actors that had a registration date earlier than their birth date or a membership cancellation date before their registration date.

Lastly, all actors whose memberships had already been terminated before 2013 were removed from the data. The remaining data consists of around 200.000 actors and the previously mentioned 10 columns.

2.3.3. Neighbourhood data

For this research, it is not known if any of the neighbourhood data features could be useful in predicting whether or not a member will terminate their membership in the coming half year, so the obvious choice is to keep all of these columns. However, it seemed highly unlikely that the average distance to any facility other than the nearest library would be of influence, so all other 58 distance columns were removed.

2.3.4. Normalization of Numeric Features

For most models, the numerical data needs to be normalized, which means that the mean of the data should be zero and the standard deviation of the data is set to 1. This normalization needs to be done for each feature that contains numeric variables. In the Appendix, it is noted which columns contain numeric and which columns contain categorical data.

2.3.5. Dummy Variables for Categorical Features

Most models cannot handle categorical variables with more than 2 possible levels, or can at most handle these in a very slow and inefficient manner (Hastie, Tibshirani, & Friedman, 2009).

Some random forest implementation will notoriously attempt every partition of levels to calculate the best split, resulting in enormous calculation times (Breiman, 2001).

Therefore, it is common practice to use dummy variables instead. This means that for each categorical feature, N new features are made, where N is the number of different values that this categorical feature can have. We next assign the value 0 to each column that does not match with the value of the original feature and we assign the value 1 to the column that does match the original feature. This process is repeated for each categorical variable.

2.3.6. Handling Missing values

All data provided by BiSC that is left over at this point does not contain any missing values. However, as mentioned in section 2.1.4, the neighbourhood data does contain some missing values for certain neighbourhoods. The models used in this research are able to handle missing values, but not all machine learning algorithms are. There are many ways to deal with missing values in features. The most common one being imputation, where the missing value is replaced by some value, generally based on the known values of that feature. The most common imputation methods insert values from random different entries, known as hot-check, or the mean value of the known entries (Enders, 2010).

However, since neighbourhood data is very specific and bound to location, and some of the columns have specific properties, like the different age groups adding up to 1, as well as the used models' ability to handle missing values, it was decided to leave the missing values as they were.

2.4. Experimental setup

As discussed earlier, the goal of the research was to predict if customers would terminate their library memberships, based on their loaning behaviour. In order to make this prediction, some decisions had to be made on how this prediction would be done. Firstly, a choice needed to be made between a regression model that predicts the expected remaining lifespan of a customer and a classification model that predicts whether a customer will leave the library within some timespan.

Secondly, if the last of those would be chosen, a fitting timespan would have to be decided on. For example, if the timespan is too long, the prediction will not be of much use to the libraries. But if it is too short, there will be no time to act on the prediction. After some discussion with

BISC employees, it was decided that the model would predict whether a customer would end their membership in the next six months. This process has been captured in the research question that was presented in the introduction.

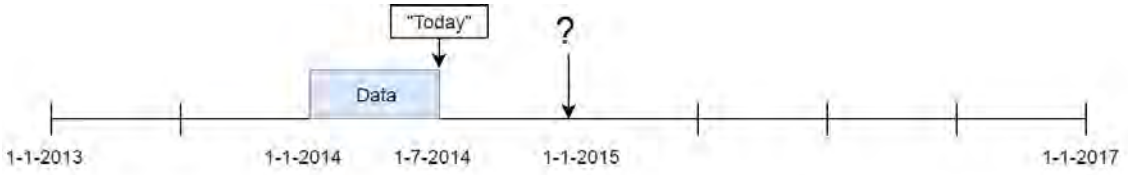

Figure 7: Graphical representation of the data setup

In order to get as many data points as possible using this framework, it was decided to divide the data into 6-month time periods. These eight groups can then be used to form seven pairs that each describe a full year of data. For each of these groups, the first half year is used as input, while the second half year is used to create the dependent variable. In Figure 7, we see this setup used for the third of these groups. The third half year of data will be aggregated to form the features, while we look at the end of the fourth half year for the value of our dependent variable.

For each of the seven time groups, if a member had an active membership at the end of that time period, that member would be included in the entries for that time period. Then, for each member that was included in a time period, the target variable could be determined from the following time period; the target variable is 1 if the member still has a membership at the end of the next time period and 0 otherwise. At this point, the neighbourhood data for the members can be added by matching the neighbourhood data and the customer data on the ZIP code. Finally, the loan data for this member in this time period can be aggregated and additional features can be created.

Creating Additional Features

Besides the total number of loans in this time period, more features were created that could have predictive power. All features based on the loan data were as follows:

- Number of loans
- Number of visits
- Time since last visit
- Percentage of loans at the membership registration location
- Mean, standard deviation, minimum and maximum of loans per visit
- Mean, standard deviation, minimum and maximum time between visits

In these features, a visit is defined as a group of loans that took place on a single day at a single location. When a member physically visited a location, but didn't loan any books, it is not regarded as a visit in this model. The loan-based features were created by separating all loans first by customer and then, for the visit features, by date and then doing the calculations for each member.

After aggregating all the datasets for all seven groups and adding all additional features, the final dataset now consists of 1.1 million entries of 218 columns each. Of these entries, about 75 thousand have a target value of "true", indicating the cancellation of a membership in the coming six months, which is about 6% of all entries. As a final note, before adding the dummy variables, there were 80 columns, but due to some categorical features having many levels, the number of columns is more than doubled in the process.

3. Models

In order to decide which models were to be used to perform this prediction task, related works were studied. In 2004, (Poel & Larivière) conducted a massive research into churn analysis in financial services, using various hazard models. They found the importance of many specific features that significantly contributed to the churn rate in customers. These variables do not coincide with variables found in libraries; however, they define variable groups that are expected to have predictive quality in churn analytics. These groups are customer behaviour, customer demographics, macro environment and customer perception.

Although this research was based on survival analysis, there are many papers describing churn analytics using machine learning in the financial sector. Models used in these papers include Decision Trees (Prasad & Madhavi, 2012), Random forests (Burez & Poel, 2009), SVM (He, Yong, Wan, & Zhao, 2014) and Gradient Boosting (Burez & Poel, 2009). All of these models yielded good results on their respective training data. It must be noted that this does not guarantee any success in the current research, as the library environment is very different from the financial sector, as the purpose of the library is providing social services and focussing on creating as much social value as possible, where the financial sector is focussed on creating as much profit and monetary value. However, there are enough similarities, namely the availability of customer and behavioural data and the need to predict churn, to base the models used in this project on these papers.

In this chapter, we will first take a look at the theoretical background of the models. Next, we will define some performance measures with which we can compare the performance of the different models. We will also define some benchmark prediction models which will help in setting a performance baseline for all models. Finally, we will discuss the parameter optimization techniques that were used.

3.1. Model Background

In this section, the models used in this researched will be described in detail. These models are Decision Trees, Random Forest and the Extreme Gradient Boosting model. We will go over their definitions, their parameters and, if needed, some mathematical background. The statistical software program R was used for all models.

3.1.1. Decision Tree

A decision tree classifier is one of the most basic and often-used machine learning models in (binary) classification problems. In a decision tree prediction, new entries travel through the tree by comparing the value of a single feature to the decision boundary that was learned by the model in every node. The top node of the tree is generally referred to as the root of the tree. The root node is the parent node of the two child nodes below it. A subtree of a child node and its children can be referred to as a branch. Each subsequent node has child nodes of its own. A node that does not have children is called a terminal node or a leaf. Eventually the entry ends up in one of the bottom nodes, called leaves, and is assigned a class. In Figure 8, an example of a decision tree can be seen. The values in the terminal nodes, R_i, correspond to classes in the case of prediction trees, or to continuous values in case of a regression tree. In case of a binary prediction, the output is generally a value between zero and one, allowing for the user to set different decision thresholds for the prediction. Generally a value of 0.5 is used as a threshold, where predictions over 0.5 are given a "true" classifications and predictions below 0.5 are given the "false" classifications. Changing this decision threshold will influence the performance of the model.

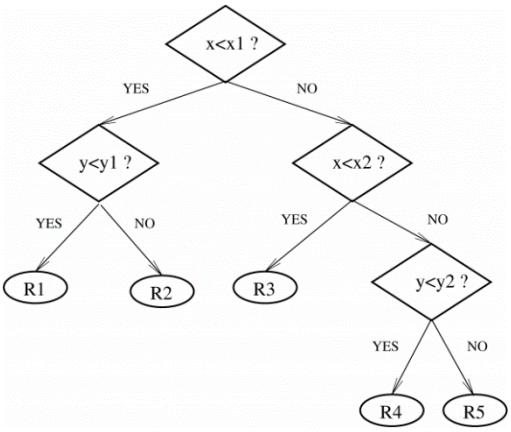

Figure 8: An example of a binary decision tree. (Source: otexts.org/1512)

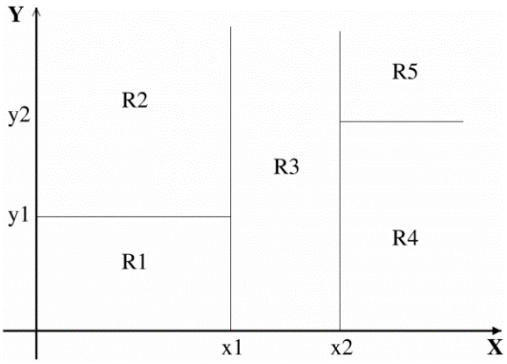

Figure 8: The resulting partitioning of the input space from the Tree in Figure 8 (Source: otexts.org/1512)

Decision trees are built by repeatedly splitting nodes on the feature with the highest information gain until all terminal nodes meet one of the predetermined stop criteria. Generally, these criteria entail that either the maximum depth of the tree has been reached, or there are fewer entries in a node than the minimum number required for a split.

There are many ways to construct node splits in tree models, but the most commonly used algorithms use the information gain metric, which is based on the concept of entropy. It is defined as follows for a general classification problem with *n* classes (Witten, Frank, & Hall, 2011):

$$E(p_1, p_2, \dots, p_n) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 \dots - p_n \log_2 p_n$$

where E is the entropy and p_i is the fraction of class i in the node. The information gain (IG) is defined as the difference of the entropy of the parent node and the weighted sum of the entropies of all child nodes.

The strongest feature of decision trees as a machine learning model is the ability to visually represent the model. Where most other machine learning models are generally black box-models, where it is impossible to visually represent what is actually going on in the model, a decision tree can show you exactly which decisions the model makes to assign a class to new entries and the most important features are immediately visible. They are the features that appear in the top of the decision tree.

However, decision trees have some serious limitations. Because of their simple structure, their performance is often not as good as more sophisticated models. Furthermore, decision trees are not very robust: a small change in the training data can have drastic effects on the resulting model (James, 2013). Moreover, without limitations of the tree size, decision trees can grow into very large, overly complex models that are too specific for the training data, leading to overfitting. Isn't this a problem for all models? There are some techniques to avoid this, however (Bramer, 2007). Maybe again mention that decision trees yielded good results in earlier studies in churn analytics, and that's why you are going to use them.

3.1.2. Random Forest

The initial algorithm for random forests was described in (Ho, 1995), but it was eventually extended and even trademarked by Leo Breiman. He defines a random forest as follows: "A random forest is a classifier consisting of a collection of tree-structured classifiers" and "each tree casts a unit vote for the most popular class" (Breiman, 2001). In other words, a large number of decision trees are constructed simultaneously, after which the class prediction is done by means of a voting system, where each tree can cast a vote.

Breiman showed that adding more trees to the model will not cause overfitting, but the generalization error does converge, so eventually adding more trees will only slow down the algorithm with little to no improvement to model performance.

In order to create variance in the decision trees, feature selection for splits is generally randomized in each node (Dietterich, 1998). Breiman (2001) finds that this method of choosing splits works very well for classification, but not as well for regression. Because this research concerns a binary classification problem, this is beneficial.

There are multiple implementations of the random forest algorithm available in R (Modi, 2016), and after some personal research, as well as recommendations from fellow data scientists, the Ranger package for R was used in this research.

This particular random forest implementation has many configurable parameters, and these are the most important ones:

- *num.trees* – The number of trees that are used. As mentioned before, increasing this number will not cause overfitting, but there is a bound on the performance of this model, and increasing the number of trees will result in a longer calculation time, so it should not be too high.
- *mtry* – The number of features to possibly split at in each node. Increasing this number will increase the search space for each tree. A suggested value is the rounded square root of the number of features, which generally balances quality and calculation time.
- *min.node.size* – The minimal node size. Decides when splits stop occurring in nodes. The smaller this number, the more precise each tree gets, but this can lead to overfitting in single trees.

3.1.3. Extreme Gradient Boosting

The Extreme Gradient Boosting is an extension of the more general Gradient Boosting model that was first introduced in (Friedman, 2002). This model is an additive composite model of decision trees, where each new tree that is added is fitted to the errors that are left from the previous ensemble of models. By repeatedly fitting a new model to the errors of the previous ensemble, the predicting power of the model is optimized. In order to improve the model, for each tree, like with random forests, the data on which the tree is based is randomly sampled from the training set. This method both speeds up the model, and prevents it from overfitting on the training data. An example of this process can be seen in Figure 10. Finally, each added model is multiplied by a learning rate variable between 0 and 1, improving generalization (Hastie, Tibshirani, & Friedman, 2009).

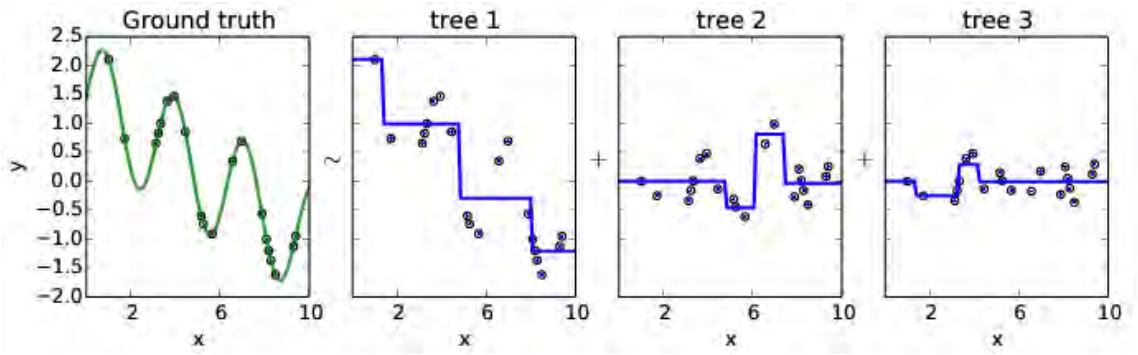

Figure 9: An example of how Gradient Boosting works. In green the original function the data points were taken from. Next the first tree prediction in the second image. In the third image, a new tree is fitted to the errors from the first tree. This process is repeated. (source: quora.com)

The Extreme Gradient Boosting algorithm is a more efficient approach and implementation to this model that was created by (Chen & Guestrin, 2016). Some improvements over earlier gradient boosting algorithms are that it allows for automatic parallel computation and multiple objective functions and allows custom objective functions to be defined. Because this is a more extensive implementation, there are many different variables that can be customized. However, many of them already have very good default values. For some of the more important ones, here are some descriptions:

- *Nrounds* – The number of iterations
- *Eta* – The model's learning rate, or the contribution of each individual tree to each model. This should have a value between zero and one. The proposed default value is 0.3
- *max depth* – The maximum depth of the individual decision trees used in the algorithm. According to Hastie et al. (2009), a value between 4 and 8 is advised. Smaller values will not guarantee enough specificity and values higher than 10 should never be required.
- *Gamma* – The minimum loss reduction required to make a further partition on a leaf node of the tree. The larger its value, the more conservative the algorithm will be.

3.2. Performance measures

In order to compare the performance of the models, some performance measures need to be defined. In this section, the following performance measures will be defined:

- Accuracy
- Precision
- Recall
- F_1 and F_β – scores

For this research, the F-scores give the best balance between precision and recall. Therefore it will be the main performance measure with which the models will be compared to each other.

In case of binary prediction, and given the observed values of a test set and the predictions of a model, these values can be displayed in a contingency table or confusion matrix. This can be displayed as follows:

		Observed class	
		True	False
Predicted Class	True	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

Table 1: Confusion matrix for a binary classification problem

Based on the number of observations that fall in each of the four categories, the following performance measures can be defined (Olson & Delen, 2008):

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Observations}$$

$$Precision = \frac{True\ Positives}{True\ Predictions}$$

$$Recall = \frac{True\ Positives}{True\ Observations}$$

An optimal classification model will have a value of 100% for all three of these measures. Using any one of these as a defining measure is generally a bad idea, without really looking in to how the values are formed (Powers, 2007). For example, when the data is severely biased to false observations, it is easy to obtain a high accuracy by always predicting false. However, this will result in a precision and recall of zero.

One way to circumvent this problem is to find a healthy balance between precision and recall by using the F-score. The traditional F_1-score or Dice Similarity Coefficient was proposed in (Dice, 1945). In this performance measure, Precision and Recall are weighed equally as follows:

$$F_1 = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

This formula can be generalized to a performance statistic that can be generalized to prefer either precision or recall, which is the F_β-score. This score is calculated as follows:

$$F_\beta = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall}$$

For the F_β-score, the user thinks that recall is β times as important as precision (Van Rijsbergen, 1979). As a result, β is a positive real number. Since we are more interested in finding the people with a high probability of ending their membership, it would make sense to use a F_2-score or perhaps an even higher value of β.

It must be noted that these performance measures all heavily depend on the positives: true negatives are not taken into account, except for the accuracy. Because of the nature of the research, this is acceptable. Through discussion with BiSC employees, it became apparent that approaching members that are not considering cancelling their membership is not detrimental, as long as the way they are approached is appropriate to their demographic features and their needs. However, when negatives do have a certain importance, there are probably better performance measures to use, like the Informedness or the Kappa score (Powers, 2007).

3.3. Benchmarks

In order to test the performance of machine learning models, we cannot simply rely on standard performance metrics such as the accuracy or recall of the model. To truly find the added value of using a model over simple guesswork, we will need to see what the performance of this "simple guesswork" really is. In this context, however, "simple guesswork" may not be as straightforward as one might think.

In general, for binary decision problems, a good benchmark would be a random guess between true or false. Assuming we know the distribution of true/false entries in the training set, we can use that ratio as our distribution of true/false guesses on the test set. Another notable benchmark would be to always predict either a positive or a negative value for any new

observations. Finally, we construct a final benchmark based on topical knowledge. By discussing the subject of benchmarks with the BiSC employees, the expectation was risen that people who do not actively loan books would have a high probability of ending their membership. This would make for a simple, easy to calculate benchmark, namely predict true if the number of loans in the past half year is equal to zero, and false otherwise. More formally:

$$Total\ number\ of\ loans = 0 \begin{cases} True: Predict\ 1 \\ False: Predict\ 0 \end{cases}$$

As an example, given a random input set, we can expect the following predictions by the benchmarks:

…	#loans	…
	5	
	0	
	3	
	0	
	1	

Always True	Always False	Random	Loans Based
1	0	0	0
1	0	1	1
1	0	0	0
1	0	0	1
1	0	0	0

Table 2: Left table denotes the input with the loans table, right table denotes the benchmark prediction. As a note, the random prediction has the same true/false ratio as the train data and the loan prediction is 1 exactly where loans is 0

3.4. Parameter optimization

As noted in Section 3, the algorithms contain several parameters that need to be set. However, for most parameters, there are strong guidelines as to what their value should be. For example, with decision trees, their depth should be around four to six layers (Hastie, Tibshirani, & Friedman, 2009). This range is small enough to test in a brute-force manner. As for the minimal number of entries in terminal nodes, this could be more complicated; however, the number of entries in terminal nodes exceeded the recommended amount by a large margin, so this value could be kept at a default.

The only algorithm that needed a more sophisticated manner of parameter optimization is the Extreme Gradient Boosting algorithm. There were too many value combinations to test by hand, so a grid search implementation was used.

A grid search is an automated way of finding the optimal parameter settings for a group of parameter ranges. First, you specify which parameters you want to optimize, the parameter ranges and their increments. The algorithm will then automatically create models for each

combination of parameters, record their performance and return the model and parameter settings of the best model. The performance criterion should also be defined beforehand. In this way, the best parameter settings can be found without much human effort. However, when searching through a large number of different combinations, calculation time will grow rapidly, and adding more parameters will make calculation time grow exponentially.

For the extreme gradient boosting model, the following variables and variable ranges were searched using this grid search principle:

Variable Name	Min	Max	Step
nrounds	800	1200	100
beta	.01	.1	.01
maxdepth	4	8	1
gamma	1	2	.1

Table 3: Variable ranges that were considered in the grid search algorithm for the XGB model.

4. Results

In this section we report the best results that were found with the different models. The models all show a similar outcome, where they are very good in predicting which customers are at risk of ending their membership, giving a high recall. However, their precision is all very low. These results are not undesirable for this specific purpose.

4.1. Decision Tree

The decision tree model has the overall lowest performance of all the models tested with an F_2-score of 0.426. This score is based on a recall of 87.8% and a precision of 13.9%. The overall accuracy of this model is 65.3%.

As mentioned in Chapter 3, we should be careful not to draw direct conclusions from the model. In the top nodes of different trees that were made, we generally see variables that are related to the loans, such as max loans per visit, total loans or time since last loan, indicating that the lack of loaning is the strongest separating feature. This does not indicate causality between these features and membership cancellations, but it does indicate correlation.

4.2. Random Forest

The random forest model outperforms the decision tree model by a large margin, reaching an F_2-score of 0.526. The recall is nearly identical to that of the decision tree model at 88.2%, but its precision is considerably higher at 20.1%. As a result, the accuracy of this model is also higher, at 77%.

In order to see which features add the most predictive value to this model, we can look at the so-called feature importance. In random forests, the importance of a feature is calculated by removing the feature from the dataset and again computing the model. If a feature is important for prediction, removing the feature from the model will lead to a decrease in performance. The larger the decrease, the more important the variable is. This method was first proposed in (Breiman, 2001). The variable importance only tells us what the predictive value of a variable is, but it does not show which values of this variable correlate with positive or negative predictions. This is one of the inherent flaws of black-box machine learning models.

For the model used in this project, the variables with the highest importance can be found in Figure 11. In this figure, we see some of the features extracted from the loan data, among which “gem_uitl_per_dag’ and “max_uitl_per_dag”, the average loans per visit and the maximum loans per visit. The feature “laatste”, which denotes the time since the last loan, is also high up

the list. The top entry is surprising, as it is the birth date of the member, which is a static date, and not related to the moment in time the data was based on. This could mean that some

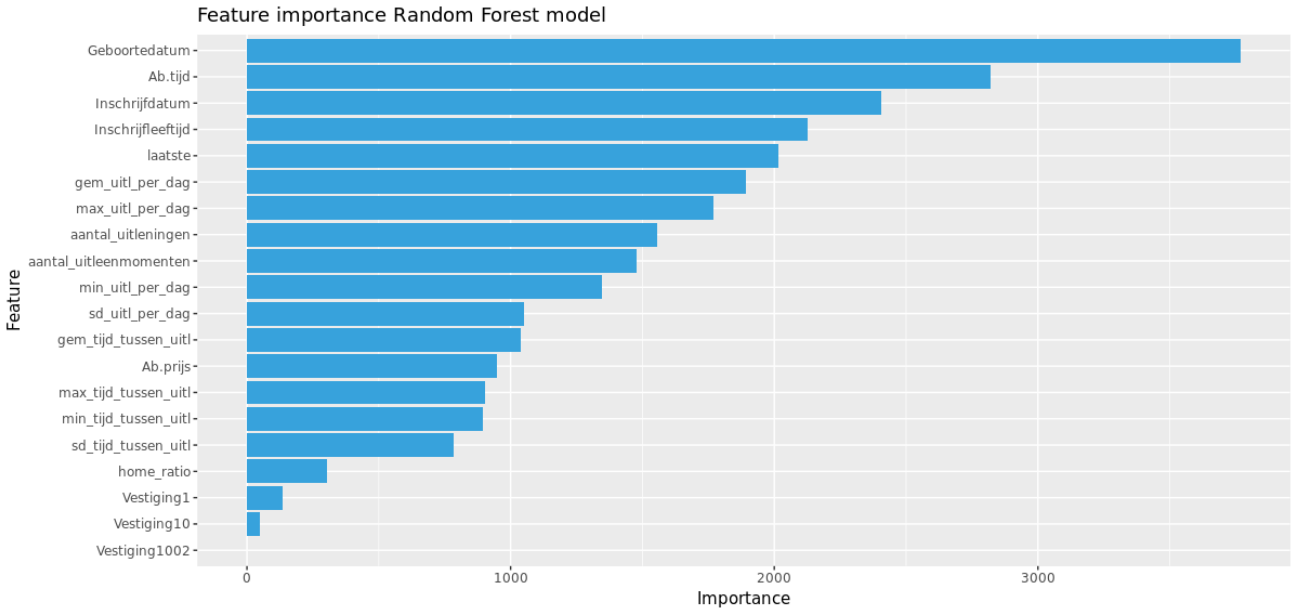

Figure 10: Feature importance of the best found random forest model

birthdays or maybe some range of birthdays has a very high predictive value for this model. The same holds for the registration date ("inschrijfdatum"). Below this, we find more intuitive features, such as the membership duration ("ab.tijd"), and the age at the start of the membership ("inschrijfleeftijd").

In general, the features with high importance are in accordance to the initial expectations that were raised by BiSC at the start of the research, except for the top feature.

4.3. Extreme Gradient Boosting

The best Extreme Gradient Boosting model that was found was the result of a grid search in the hyperparameters, as discussed earlier. The optimal values of the parameters were as follows:

- Nrounds = 1000
- beta = .03
- max depth = 6
- gamma = 1.5

This model is also the best performing model across all models that were considered, achieving an F_2-score of 0.568. The Extreme Gradient Boosting model performs better than both the Decision Tree and the Random Forest on all criteria. The precision of the best model was still relatively low, namely 22.6%, but it is still the highest found precision in this research. The recall

of this model is 91.4% and the accuracy is also slightly higher than that of the random forest at 80.3%.

Like with random forests, we can construct feature importance for the extreme gradient boosting model in a similar way. The most important features according to the XGB model can be found in Figure 12. As with the random forest model, we see that the "Geboortedatum" variable has the highest importance. This is interesting, as it is a given date, and not the time that the membership is active ("ab.tijd"). This "ab.tijd"-feature is also among the highest-importance variables, but somehow the birth date is more important. We also see some membership descriptions that appear in this graph, namely some of the standard membership and some of the youth memberships. This is not particularly interesting, since most members fall into these groups. Finally we do see the variables "aantal_uitleningen" and

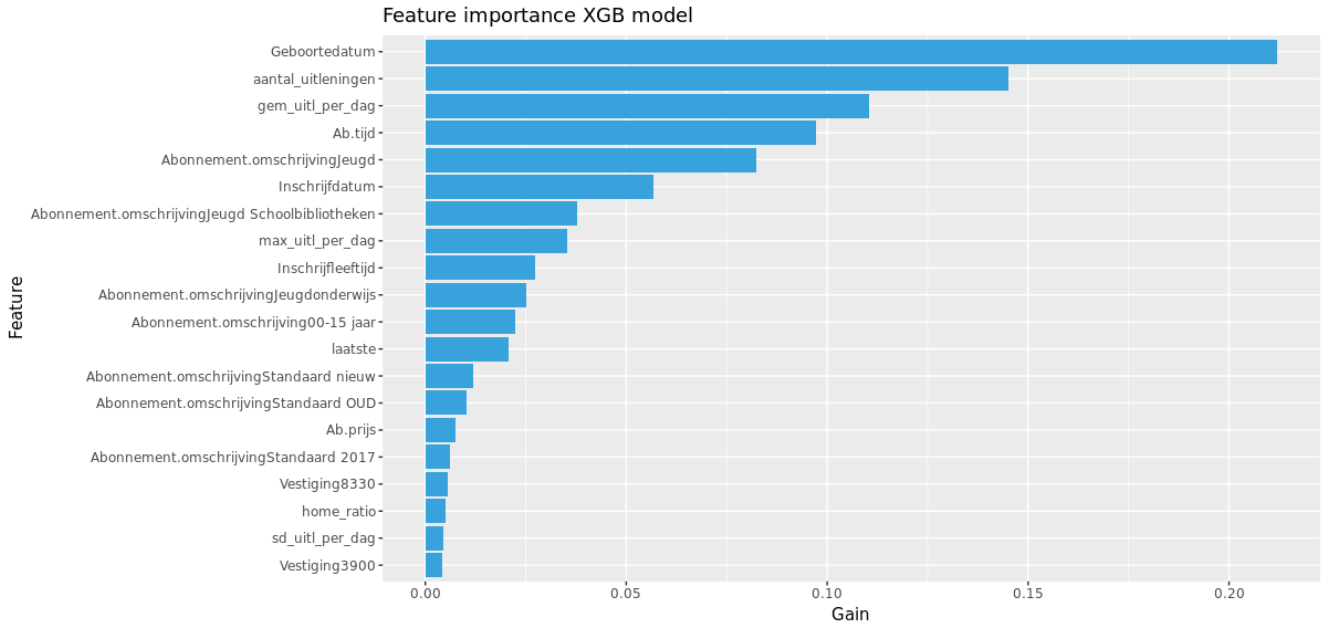

Figure 11: Feature importance of the best found XGB model

"gem_uitl_per_dag", which are the total number of loans and the average number of loans per visit, respectively, at the top of the list. This is in accordance to the benchmark model that was suggested by the BiSC employees. Again, it must be noted that we do not know whether high or low values of these features cause a positive prediction, so these importances must always be interpreted carefully.

As a final remark regarding the important variables, we see that none of the variables from the neighbourhood data can be seen in this list. This could mean that either these data really is of no importance and doesn't have any predictive value, or maybe the neighbourhood level is just

not specific enough and more detailed data is needed, perhaps on a household level. This data, of course, is much more sensitive and is not readily available.

Comparing the model to the benchmarks, we see that it greatly outperforms the trivial benchmarks, but it also improves the intuitive benchmark by a wide margin, as that benchmark only achieves an F_2-score of 0.384. In Table 2, a complete overview of all the performance measures for all models can be found.

	Accuracy	Recall	Precision	F_2-score
Dec. Tree	65.3%	87.8%	13.9%	.426
Random Forest	77%	88.2%	20.1%	.526
XGB	80.3%	91.4%	22.6%	.568
All True	6.2%	100%	6.2%	.24
All False	93%	0%	0%	0
Random	88.2%	5.5%	5.7%	.004
"Activity"	74%	14%	69%	.384

Table 4: Performance measures for all models and the different benchmarks

5. Conclusion

The best performing model is the Extreme Gradient Boosting model. It reaches a recall of 90% and outperforms the benchmark predictions by a wide margin. It must be noted that all models outperform the intuitive benchmark prediction. However, this is the intuitive approach that would have been used by BiSC due to the lack of previous research in this field.

The models considered in this project can predict which members have a high risk of ending their membership the coming six months with a very high recall, while maintaining a decent level of accuracy. The precision of these predictions is relatively low. Since high recall and low precision are shared by all models that were tested in this research, the expectation is that the data is the main cause of this. This could mean that either the data itself is the reason that it is difficult to precisely predict the positives, or the way the data has been pre-processed is the reason. The choice was made to not include some columns from the different datasets, and missing values in the neighbourhood data were not imputed. Maybe with different pre-processing techniques, some patterns in the data could be explained by the machine learning models where they could not be with the current approach. Several suggestions for improvements can be found in the recommendations (Chapter 6).

When looking at the importance of the different features for predicting membership termination, most features are in accordance to the initial expectations, being that youth members, as well as the number of loans are of importance to the churn pattern. However, also the birth date of the member, membership duration and age at the start of the membership are of importance. It must be noted that the feature importance does not tell us which values of these features lead to a positive or negative prediction. Regression analysis of the features could lead to insights on these relations, but it is likely that correlation of multiple variables makes this impossible. For example, for some feature a high value will lead to a positive prediction in combination with one different feature, but it can lead to a negative prediction in combination with another.

When looking at all these factors as a whole, it seems that even though the performance of the resulting model is good, it will be difficult to implement it directly in order to reduce churn. On the one hand, the model showed variables that were known risk factors, while the previously unknown variables did not have any logical interpretation. Adding the impossibility to find out which value ranges of these variables lead to predictions due to the black box principle of the

model, it is not directly possible to indicate risk groups within the customer base. In short, the model cannot be implemented in its current state.

6. Recommendations

As the results show, using the data that was available for this research, the resulting XGB model can already make predictions with a high recall. At the present day, more data is available, so retraining the model with all available data up until today and regularly updating it should increase performance even more over time.

In the data preparation step, certain choices for methods were made on intuition, or unfamiliarity with other techniques at that time. For future research, more techniques could be tested where these choices were made. For example, the unbalance of the data was solved by down-sampling the negatives, while there are other more sophisticated methods of balancing the dataset that could potentially yield better results. Also, not all data was taken into account. In all datasets, certain columns were not used in the machine learning model. Perhaps there are still patterns in this omitted data, even though this goes against all expectations.

A number of additional features were created for this research, but perhaps there are still some features that could be constructed which have sufficient predictive value. The general experimental setup which divided the available data into seven sets of a year was thought to be a good approach, but perhaps a different approach to this problem would yield better results. Perhaps a regression model or survival analysis could lead to a model that has a superior performance to the model that was created in this research.

The neighbourhood data provided by the CBS was not complete for every neighbourhood, leading to some missing values. Even though the models used in the research can handle missing values, it could be beneficial to see if this missing data can be gathered or, if not, be imputed, even if this brings difficulties. If so, more models can be tested that would not work correctly with missing values. This being said, the neighbourhood data did not appear to add any value to the models, so alternatively it could be recommended to not use this data in any further research.

7. Bibliography

Bramer, M. A. (2007). *Principles of Data Mining.* London: Springer.

Breiman, L. (2001). Random Forests. *Machine Learning*, 5-32.

Burez, J., & Poel, D. V. (2009). Handling Class imbalance in customer churn prediction. *Expert System With Applications*, 4626-4636.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System.

Dice, L. R. (1945). Measures of the Amount of Ecologic Association Between Species. *Ecology*, 297-302.

Dietterich, T. (1998). An experimental comparison of three methods for constructing ensembles of decision trees. *Machine Learning*, 1-22.

Enders, C. (2010). *Applied missing data analysis.* New York: Guilford Press.

Friedman, J. (2002). Stochastic Gradient Boosting. *Computational Statistics & Data Analytics*, 367-378.

Galetti, M. (2015, July 9). *What is attrition rate?* Opgehaald van ngdata.com: https://www.ngdata.com/what-is-attrition-rate/

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning.* New York: Springer.

He, B., Yong, S., Wan, Q., & Zhao, X. (2014). Prediction of customer attrition of commercial banks based on SVM model. *Procedia Computer Science*, 432-430.

Ho, T. K. (1995). Random Decision Forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, (pp. 278-282). Montreal.

James, G. (2013). *An Introduction to Statistical Learning.* New York: Springer.

Modi, M. (2016, December 13). *Different random forest packages in R*. Opgehaald van linkedin: https://www.linkedin.com/pulse/different-random-forest-packages-r-madhur-modi

Olson, D. L., & Delen, D. (2008). *Advanced Data Mining Techniques.* Springer.

Poel, D. v., & Larivière, B. (2004). Customer attrition analysis for financial services using proportional hazard models. *European Journal of Operational Research*, 196-217.

Powers, D. M. (2007). *Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation.*

Prasad, D., & Madhavi, S. (2012). Prediction of churn behavior of bank customer customers using data mining tools. *Business Intelligence Journal*, 96-101.

Van Rijsbergen, C. J. (1979). *Information Retrieval.* Butterworth.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining.* Burlington: Morgan Kaufmann.

A. Appendix

Column Names Actor Data

Name	Interpretation	Type[2]
Actor.id	Index	C
Vestiging	Location Member is Registered to	C
Geboortedatum	Date of Birth	D
Sexe	Sex	C
PC.6	ZIP Code	C
Lid.status	Member Status	C
Abonnement.omschrijving	Membership Type	C
Stat..Ledencategorie	Member Category	C
Inschrijfdatum	Registration Date	D
Uitschrijfdatum	Cancellation Date	D
Rol	Role	C
Klant.in.regio	Living Region of Member	C
Extra.rol	Additional Role	C
Samengestelde.rol.	Compound Role	C
Einddatum.rol	Role End Date	D
Abonnement..binnen regio.	Combination of Region and Membership Type	C
Ab.prijs	Membership Fee	N
Ab.periode	Membership Duration	C

Column Names Loan Data

Name	Interpretation	Type
Instantie_id	Location ID	C
punt	Loan Terminal ID	C
bron	Loan Source	C
aktie_datum	Data of Loan	D
aktie	Loan/Return/Reloan Indicator	C
materiaal	Media Type	C
scatnr	Media Category	C
titelnr	Collection ID	C
actor_id	Member ID	C
aktie_vest	Location ID	C

[2] N = Numeric, D = Date, C = Categorical

Column Names Collection Data

As mentioned in Section 2.1.3, The interpretation of some of these columns is extremely vague. Since this dataset was not used extensively in the research, there was no effort made to try and find all of the missing interpretations, as they were not needed.

Name	Interpretation	Type
exem_id	Item ID	C
titelnr	Title ID	C
actor_id	ID of actor that last loaned	C
zebra	Zebra ID	C
stamboeknr	ID of Reference Book	C
eigenaar	ID of Owner (library establishment)	C
oorsprong	ID of Origin	C
uitvest	Location	C
doelvest	Location	C
etiket_scat		C
materiaal	Type of Media	C
matvolume	Number of parts	C
plaatsing	Location within library	C
pldetail		C
plaatsopm		C
status	Status	C
substat	Secondary Status	C
aantal		N
aantalj		N
aantalv		N
leen_dat	Loan Date	D
inl_dat	Return Date	D
vin_dat		D
maantel		C
bericht_type		C
bericht_datum		C
verltel		C
volgnr		C
bijlage		C
alg_blok		C
res_blok		C
ver_blok		C
akt_blok		C
sort_nr		C
prijs		N
best_dat		D
best_nr		C

ontv_dat		D
herkomst		C
nbd_cat		C
leen_blk		C
leen_blk_datum		D
kast_nr		C
period_per		C
period_jrg		C
opberg_nr		C
jeugd_blok		C
verl_dat		D
verl_bron		C
leen_geld		N
nivo		C
hwoord		C
taal		C
genre		C
leveranc		C
bestabon		C
opm_etik		C
magazijn		C
blok_publ		C
balie_mld		C
kast_eign		C
inven_dat		D
schade		C
min_lft		C
exm_titel		C
exm_auteur		C
exm_kast		C
exm_kastpub		C
exm_etiket		C
kast_temp		C
kast_datum		C
balie_bits		C
hervzm_dat		D
ggc_ppn		C
hoofd_exem_id		C
revisie		C
afschrijf_blok		C
opruimtijd		D
updated		D
bindwijze		C

Column names Neighbourhood Data

Variable Name	Interpretation	Type
bu_code	Neighbourhood index	C
bu_naam	Neighbourhood Name	C
wk_code	District Index	C
gm_code	Municipality Index	C
gm_naam	Municipality Name	C
ind_wbi	Indicator if data changed since last year	C
water	Large Body of Water Present	C
postcode	ZIP Code	C
dek_perc	Similarity of ZIP code in Neighbourhood	C
oad	Number of Addresses per km^2	N
sted	Urbanity Index	C
aant_inw	Number of Inhabitants	N
aant_man	Number of Men	N
aant_vrouw	Number of Women	N
p_00_14_jr	Percentage of People age 0-14	N
p_15_24_jr	… 15-24	N
p_25_44_jr	… 25-44	N
p_45_64_jr	… 45-64	N
p_65_eo_jr	… 65 and over	N
p_ongehuwd	Percentage of Unmarried People	N
p_gehuwd	… Married People	N
p_gescheid	… Divorced People	N
p_verweduw	… Widowed People	N
geboo_tot	Total number of births	N
p_geboo	Births per 1000 Inhabitants	N
sterft_tot	Total number of deaths	N
p_sterft	Deaths per 1000 Inhabitants	N
bev_dichth	Population Density	N
aantal_hh	Number of Households	N
p_eenp_hh	Number of Single Person Households	N
p_hh_z_k	Number of Households without Children	N
p_hh_m_k	Number of Households with Children	N
gem_hh_gr	Average Household Size	N
p_west_al	Immigrants per 1000: Western Countries	N
p_n_m_al	… Non-Western Countries	N
p_marokko	… Morocco	N
p_ant_aruba	… Netherlands Antilles and Aruba	N
p_surinam	… Suriname	N
p_turkije	… Turkey	N
p_over_nw	… Other	N
a_bed_a	Number of Companies: Agriculture	N
a_bed_bf	… Industrial	N
a_bed_gi	… Trade and Catering	N
a_bed_hj	… Transport, Information and Communication	N
a_bed_kl	… Financial Services and Real Estate	N

a_bed_mn	… Business Services	N
a_bed_ru	… Culture, Recreation and Other	N
woningen	Number of Houses	N
woz	Average House Value	N
p_1gezw	Percentage of Single-family Houses	N
p_mgezw	Percantage of Multi-family Houses	N
p_wont2000	Percentage of Houses built before 2000	N
p_wonv2000	Percentage of Houses built after 2000	N
auto_tot	Number of Cars	N
auto_hh	Number of Cars per Household	N
auto_land	Number of Cars by Land	N
bedr_auto	Number of Company Vehicles	N
motor_2w	Number of Motorbikes	N
a_lftj6j	Number of Cars <6 Years Old	N
a_lfto6j	Number of Cars >= 6 Years Old	N
a_bst_b	Number of Cars (gasoline)	N
a_bst_nb	Number of Cars (no Gasoline)	N
af_ziek_i[3]	Average distance to nearest Hospital	N
af_ziek_e[3]	… Hospital	N
af_superm[4]	… Supermarket	N
af_warenh[3]	… Department Store	N
af_cafe[4]	… Café	N
af_caftar[4]	… Cafetaria	N
af_restau[4]	… Restaurant	N
af_hotel[3]	… Hotel	N
af_kdv[4]	… Children Daycare	N
af_bso[4]	… After-School Care	N
af_brandw	… Fire Department	N
af_oprith	… Highway Access	N
af_treinst	… Train Station	N
af_overst	… Public Transport	N
af_zwemb	… Swimming Pool	N
af_ijsbaan	… Ice Rink	N
af_biblio	… Library	N
af_bios[3]	… Cinema	N
af_sauna	… Sauna	N
af_zonbnk	… Tanning Salon	N
af_attrac[3]	… Amusement Park	N
opp_tot	Total surface	N
opp_land	Land Surface	N
opp_water	Water Surface	N

[3] Also contains columns of average distance to nearest 5, 10 and 20

[4] Also contains columns of average distance to nearest 3 and 5