ERASMUS UNIVERSITY ROTTERDAM

Erasmus School of Economics

Master Thesis Data Science & Marketing Analytics

Predicting CLV with supervised learning methods using music streaming subscription data

Olivier Henri Mathijs Rudolf Bax

Supervisor: dr. K Gruber

Second assessor: dr. CS Bellet

Date: 13/10/2023

Abstract

This research tried to find the most suitable supervised learning model to predict customer lifetime value. It does this by predicting the customer lifetime value of individual users by using three different types of supervised learning algorithms with and without using additional feature selection methods, using user data from a music streaming service operating in Asia. Moreover, customer lifetime value was not available in the data set and was calculated using the basic and most simple calculation method. From the nine different models that were used in this research, the RF-RFE model had the highest prediction accuracy and was chosen as the most suitable supervised learning model, despite being the least interpretable model.

Table of contents

1 Introduction

Over the last two decades, the music industry has changed a lot because of the rapid growth of the Internet (Kim et al., 2017). First, offline record sales were being replaced by online music downloading, but now, music streaming is becoming the new business model for online music services because of the emergence of mobile devices (Sugo Music Group, 2016). Music streaming services are web-based services that allow users to stream music to their phones or computers (PCMag, 2023). They earn revenue by either showing advertisements to free users or by charging a monthly subscription fee to subscribers (Wlömert & Papies, 2016). According to Susic (2023), music streaming accounts for one third of all consumption and more than 616 million people were subscribed to music streaming services worldwide in 2022.

KKBOX is the first music streaming service that provides legal music streaming services to customers (KKBOX, 2023). It offers both a limited and unlimited version to more than 10 million users in Asia and is financed by advertising and paid subscriptions (WSDM, 2018). KKBOX is currently the biggest music streaming service in Asia, featuring more than 90 million legal tracks (KKBOX, 2023). It is currently accessible in Taiwan, Japan, Singapore, and Malaysia, and remains to be a prominent Taiwanese brand that actively cultivates the music industry.

According to Jawale (2022), a subscription service should be built on customer relationship management (CRM). Hu et al. (2013) defined CRM as handling interactions between customers and organizations. The goal of CRM is to attain and maintain customers, increase their value and devotion, and create strategies that are prioritizing their needs (Cheng & Chen, 2009). A good relationship with customers is very important for commercial companies to succeed in their marketing competitions (Safari et al., 2016). One of the vital concepts of CRM is customer lifetime value (CLV), which was introduced first by Kotler (1974) more than 40 years ago. Dahana et al. (2019) defined CLV as the present value of all future cash flows from a customer.

One of the main benefits obtained from a CLV analysis is that it gives insights into the overall performance of a business and helps with making more appropriate marketing strategies and customer-related decisions (Gurău & Ranchhod, 2002). According to Blattberg et al. (2009), CLV analysis provides companies with a long-term economic view of the customer and gives insights into the key components of CLV. Furthermore, they stated that CLV combined with customer acquisition data and expenses allows firms to maintain the profitability of the firm in the long run. Moreover, Chang et al. (2012) explained that by classifying customers into high-, medium- and low-value customers, CLV models allow firms to differentiate their products and services according to the expected value of the customer. Besides that, they explained that it gives insights into how much money a company can afford to acquire new customers and to obtain the tradeoff between cost and profitability.

Calculating CLV accurately is thus an important task for businesses. However, predicting CLV with supervised learning might be an interesting next step for a company. Supervised learning has significantly increased in value and importance in recent years, as it made many things possible which were previously seen as impossible (Shah et al., 2020). Shah et al. explained that supervised learning is an attempt to create intelligent systems, while being completely based on statistics and mathematics. For example, supervised learning can be used to predict future data/unseen data by analyzing historical data/given data (Shah et al., 2020), or to detect non-linear patterns (Desai & Ouarda, 2021) in the data.

This master thesis will try to predict CLV using a least absolute shrinkage and selection operator (LASSO) regression, a regression tree (RT), and a random forest (RF), with and without using recursive feature elimination (RFE) and best subset selection (BSS), using subscription data from KKBOX. The goal of this research is to find which supervised learning model, possibly in combination with a feature selection method, is the most favorable model to predict CLV. The research question for this research is: *"What type of supervised learning model is most suitable for predicting CLV when using music streaming subscription data?"*

To answer this question, the following sub-questions will first need to be answered:

"How does the model interpretability vary for a LASSO regression, a regression tree and a random forest when predicting CLV?"

"How does the prediction accuracy vary for a LASSO regression, a regression tree and a random forest, when predicting CLV?"

"How does the prediction accuracy of each model change when recursive feature elimination or best subset selection is used in addition?"

This research is academically relevant as it builds upon existing literature on CLV prediction using supervised learning algorithms. It will provide insights how feature selection affects the accuracy of a LASSO regression, an RT, and an RF when predicting CLV. Moreover, this research uses user subscription data from a music streaming service to predict CLV, which has not been used before for this type of research.

Besides academic relevance, this research is socially relevant as the insights gained from this research have practical implications for companies. These implications can help companies to identify relevant features to predict CLV and to choose a suitable supervised learning algorithm in combination with a feature selection method when predicting CLV. Moreover, the insights can optimize marketing strategies of companies and improve customer satisfaction.

2 Literature review

2.1 What is customer lifetime value

Creating a strong relationship with customers is important for businesses to be successful in their marketing competitions (Safari et al., 2016). Kumar et al. (2008) highlighted that a firms customer management activities involves making consistent decisions over time regarding which customers to select for special targeting, resource allocation to customers and selecting customers to receive special services to increase profitability. The authors stated that it is essential to measure customer profitability and to understand the connection between firm actions and customer profitability is needed in order the make the previously mentioned decisions successful. Customer profitability often refers to the revenues minus the costs generated by a customer during a specific period, but it can also be projected into the future (Chang et al., 2012). According to Petrison & Blattberg (1997), this output is often termed as customer lifetime value (CLV). Blattberg et al. (2009) and Chang et al. (2012) gave a more detailed definition and defined CLV as the stream of expected future revenues over the lifespan of a customer, after subtracting incremental costs from production, selling, servicing, and marketing, discounted with a proper rate to get the net present value (Blattberg et al., 2009; Chang et al., 2012).

According to Chang et al. (2012), CLV serves as an efficient metric to assess a firm's customer relationships and is crucial for firms offering customer-oriented services. CLV helps businesses to understand customer value and customer patterns, which in turn allows businesses to make efficient marketing strategies, cut costs, and create long-term relationships (Chang et al., 2012; Gurău and Ranchhod, 2002). Furthermore, Chang et al. stated that CLV can assist in managing the existing customer base. For instance, categorizing customers into high, medium, and low value segments allows companies to tailor products and services based on customer value and to aim retention efforts towards high-value segments (Chang et al., 2012).

Dahana et al. (2019) explained that to accurately calculate CLV, information about the revenues gained from a customer, the cost of customer acquisition, the retention rate, the discount rate, and the lifetime duration of a customer is needed. Fader & Hardie (2007) defined the retention rate as the proportion of customers who remain active from the beginning to the end of a specific period. Moreover, Hansen & Jagannathan (1997) defined the discount rate as a value that is used to calculate prices today by discounting the corresponding cash-flows at a future date. Furthermore, Tukel & Dixit (2013) considered the lifetime of a customer to be the entire period the customer stayed in a relationship with the company.

2.2 Predicting customer lifetime value

According to Venkatakrishna et al. (2021), there are several different methods to predict CLV. Additionally, the authors stated that many studies use regression methods to predict of CLV. For example, Malthouse & Blattberg (2005) utilized different supervised learning models to predict CLV. Moreover, Glady et al. (2009A) stated that CLV can be estimated by forecasting the number of transactions and corresponding profits of a customer. Another research by Glady et al. (2009B) showed the effectiveness of probabilistic models like Pereto/NBD and BG/NBD to predict the future behavior of a customer.

What is important to note is that in this research, only the basic CLV calculation method (section 2.2.1) and the supervised learning model method (section 2.2.5) will be used. The RFM model, Pareto/NBD model, BG/NBD model and the Markov chain model (section 2.2.2, 2.2.3 and 2.2.4) are only included to illustrate alternative methods for estimating CLV. As a result, these sections provide a brief explanation. However, for a more comprehensive understanding of these models, a more detailed explanation of these models can be found in section 9.1 of the appendix.

2.2.1 Basic CLV calculation

According to Kahreh et al. (2014), the simplest way to calculate/estimate CLV is the basic formula proposed by Berger & Nasr (1998) that calculates the CLV for customer i at time t for a finite time horizon (T) in the following way:

$$CLV_{i,t} = \frac{\sum_{t=0}^{T} \pi_{i,t} * r^t}{(1+d)^t}$$

where $\pi_{i,t}$ represents the profit made from customer i in period t and d represents the discount rate to discount future cashflows/profits (Berger & Nasr, 1998). Blattberg et al. (2009) explained that these profits consist out of revenues and costs.

The model makes two major assumptions. It assumes a constant margin of profit over time that does not consider the stochastic nature of the purchase behavior of the customers and assumes that the customer behavior is uniform for all customers. Moreover, the model assumes that the customer retention rate and the costs of retention are constant over time and both costs and revenues happen periodically at a constant rate.

2.2.2 RFM model

Recency, purchase frequency and monetary value are well known metrics used in marketing (Burelli, 2019). Together, these metrics are known as RFM and are often used to predict customer behavior (Gupta et al., 2006). Shih and Liu (2003) came up with a method that uses RFM and CLV clustering to rank customers based on profitability. They explained that this is done by first identifying the relative importance of the RFM variables using analytical hierarchical processing, followed by clustering the customers based on RFM, and score these clusters using a weighted sum of the normalized RFM features. Burelli (2019) stated that this method is not able to predict numerical values for CLV, but rather to rank the customers based on profitability. Moreover, Fader et al. (2005A) further emphasized that this method is only able to predict customer behavior for the upcoming period.

2.2.3 Pareto/NBD and BG/NBD

An alternative method to predict CLV is the pareto/NBD model introduced by Schmittlein et al. (1987). The model tries to predict the number of future purchases of customers based on recency, frequency, and customer lifetime (Glady et al., 2009B). It does this by using a Pareto distribution of the second kind and a negative binomial distribution (Burelli, 2019). The Pareto distribution is controlled by the variation in customer lifetimes and the average duration of a customer's lifetime, while the negative binomial distribution is controlled by the variability in the purchase frequencies of a customer and average purchase frequency (Burelli, 2019; Schmittlein et al., 1987). These parameters can be estimated from past customer behavior and with these parameters, the model can indirectly predict CLV by predicting the number of future purchases (Glady et al., 2009B).

The Pareto/NBD model uses very complex computations (Fader et al., 2005B). As a result, the more efficient BG/NBD model was proposed by Fader et al. (2005B). Fader et al. stated that in the BG/NBD model, customer activity is modeled based on the probability of a customer making a purchase within a specific time and the probability of the customer becoming inactive after making a purchase.

2.2.4 Markov Chain Model

Pfeifer and Carraway (2000) suggested an alternative approach for predicting CLV, which involved employing a Markov Chain Model (MCM) to model the customer relationship. MCMs are mathematical models that describe random processes (Ching & Ng, 2006). Ching & Ng explained that a process is represented by a set of states, and transitions between these states are determined by probabilities. Moreover, they explained that the transitioning of each state to another state has its own associated probability, called a transition probability. Pfeifer & Carraway (2000) applied the MCM in a way where the states represented different relationship conditions between customers and the company, and the transition probabilities between the states represented the probability of a customer moving from one condition to another, for example to churn or to make a purchase.

2.2.5 Supervised learning models

According to Burelli (2019), another possible method to predict CLV would be by using supervised learning. He explained that it can be used to learn a complex function between past customer behavior and their recorded lifetime value. Supervised learning tries to predict the categorical or continuous values of a dependent variable based on input variables called features (Schrider & Kern, 2018). Burkart & Huber (2021) explained that supervised learning does this by using a training data set with labeled examples, where each example consists out of a set of features, each corresponding to a certain value of the dependent variable. Additionally, the authors explained that supervised learning learns from these examples to create a model that estimates the dependent variable for new/unseen data. When supervised learning is used for classification, the output is a discrete label and when supervised learning is used for regression, the output is a continuous value (Burkart & Huber, 2021).

Burelli (2019) explained that some supervised learning models sacrifice explanatory power for better prediction power by capturing more complex relations. These type of supervised learning models are called black-box models (Burelli, 2019). A black-box model can make relatively accurate predictions but is very hard to interpret because its decision-making process is not transparent (Zhao & Hastie, 2021). When used for CLV prediction, these models may not provide detailed insights into individual customer preferences, but they can be more accurate and can therefore be useful for tasks such as marketing automation (Burelli, 2019).

There are many different methods how supervised learning can be used to predict CLV. For example, Haenlein et al. (2007) used a decision tree and a MCM to predict CLV. They did this by first dividing their customer data of a retail bank into age groups. After that, they fitted a decision tree on each age group to predict the profit for a customer within that specific group. The authors then used a MCM to model the transition probabilities between the groups so that the model can track customer transitions throughout their lifetime. Finally, they calculated CLV by discounting the sum of predicted CLV for each possible customer states, weighted by the transition probabilities.

Moreover, Asadi & Kazerooni (2023), Sawant (2022), and Venkatakrishna et al. (2021) all employed supervised learning approaches to predict CLV that are very similar to each other. Sawant (2022) used Lasso regression, RF, and extreme gradient boosting, using demographic, educational, financial, vehicle, and policy data to predict CLV. Venkatakrishna et al. (2021) on the other hand employed linear regression, extra trees regression, RF, gradient boosting, and extreme gradient boosting, using predictors such as recency, frequency, total revenue, and transformed categorical variables to predict CLV. Moreover, Asadi & Kazerooni (2023) used stacked ensemble learning models including deep neural networks, bagging, support vector regression, RF, extreme gradient boosting, and light gradient boosting machine, using normalized recency, frequency, monetary value, average basket weight, and customer relationship length to predict CLV. Asadi & Kazerooni (2023) used a time-based split based on a threshold date to divide the data into a training and test set, while Sawant (2022) and Venkatakrishna et al. (2021) used random sampling. Furthermore, in the research of Sawant (2022), a pre-calculated CLV variable was already present in the data, while Asadi & Kazerooni (2023) and Venkatakrishna et al. (2021) used derivations of the basic CLV calculation method to calculate CLV, before predicting it using the different supervised learning methods.

2.4 LASSO regression

Regression models are often used for prediction (Ranstam & Cook, 2018). One of the best-known standard regression methods is the ordinary least squares (OLS) method, which fits a linear model using predictors to minimize the sum of squared differences between the actual values and the predicted values by a linear approximation. However, Ranstam & Cook (2018) explained that OLS often has low bias but high variance, causing the model to overfit the data. They stated that OLS works especially poorly when predicting ‘more extreme’ observations and explained that the more predictors are being used, the harder it becomes to interpret the model. LASSO regression, introduced by Tibshirani (1996), was introduced to address these problems.

LASSO is a shrinkage and feature selection method that can be used for OLS or logistic regression problems (Ranstam and Cook (2018). Tibshirani (1996) stated that shrinking coefficients of irrelevant variables towards zero can improve prediction accuracy and that using a subset of variables with the strongest effects can improve interpretability. However, the standard techniques such as subset selection and ridge regression to improve OLS estimates both have drawbacks according to Tibshirani (1996). He explained that LASSO uses the good features of best subset selection and ridge regression by shrinking some coefficients while setting others to zero.

Ranstam and Cook (2018) explained that LASSO tries to identify the variables and respective regression coefficients that result in a model that minimizes the prediction error (Ranstam & Cook, 2018). It does this by imposing a constraint (λ) that the sum of absolute values of regression coefficients should be less than a fixed value (λ), which shrinks the coefficients of irrelevant variables close to or even equal to zero (Ai, 2022). A larger λ enforces stricter constraints, which causes more shrinkage and potentially setting some coefficients to zero. The variables with coefficients equal to zero because of the shrinkage are removed from the model (Ranstam & Cook, 2018). Ranstam & Cook (2018) also explained that λ can be chosen by using k-fold cross-validation. With k-fold cross-validation, the data set is randomly split into k folds of equal size. (Anguita et al., 2012; Wong & Yeh, 2019). Anguita et al. (2012) explained that each fold in turn plays the role for testing the model while the other k - 1 folds are used for developing the model and that this procedure is carried out k times. A rule-of-thumb suggests fixing large values of k such as 5, 10 or 20 as it is often preferred to exploit many patterns to train the model (Anguita et al., 2012). Rantam & Cook (2018) explained that by combining the separate test results, the optimal λ can be chosen which can be used to determine the optimal model. Furthermore, they explained that this method reduces overfitting.

As mentioned before, two big advantages of LASSO compared to standard regression models are that it improves prediction accuracy by shrinking or setting coefficients to zero and that it improves model interpretability by only using a subset of variables with the strongest effects (Tibshirani, 1996). It even has the capability to select a subset of predictors in “sparse data” situations with many possible predictors, where only a few of them are related to the dependent variable (Roy et al., 2015). Despite the advantages of the LASSO model, it has a few drawbacks. Roy et al. (2015) stated that when predictors are highly correlated, LASSO may not generate consistent results. Moreover, Algamal & Lee (2015) stated that in the presence of high correlation among predictors, LASSO struggles to distinguish relevant and irrelevant variables Additionally, they stated that LASSO is unable to select more predictors than the number of observations in the data.

2.5 Decision tree

Decision trees (DTs) are non-linear tree like sequential models, which use a sequence of simple tests to sequentially split the data into multiple groups of homogenous data (Kotsiantis, 2013; Yang et., 2017). DTs are very easy to interpret and have become very popular because of their ability to handle covariates measured at different measurement levels (Larivière & Van den Poel, 2005; Yang et., 2017).

Loh (2011) stated that a DT can either be used for classification problems (classification tree (CT)) or regression problems (RT). Loh (2011) explained that an RT is fitted to each homogenous group of data to give a predicted value of the dependent variable, as the dependent variable takes continuous or ordered values. On the other hand, a CT splits the data into separate classes and takes categorical values for its dependent variable (Kotsiantis, 2013). The Classification and Regression Tree (CART) learning algorithm, which was introduced by Breiman et al. (1984), is the best-known decision tree learning algorithm in the literature (Yang et al., 2017).

When the CART algorithm is used for a regression problem, it looks for tests (binary divisions) that best divide the data into homogenous groups (Kotsiantis, 2013). Kotsiantis explained that every test compares a numeric variable against a threshold value and a categorical variable against a set of possible values. According to Breiman et al. (1984), Efron & Tibshirani (1991) and Venables & Ripley (1997), the algorithm starts by taking all the available training data (root node) and analyzes all possible 'splits' for every explanatory variable. The algorithm then selects the split which reduces the deviance in the dependent variable the most and this process is repeated for the two groups of data resulting from this first split, until all the groups are homogenous or until a certain criterion is satisfied (Breiman et al., 1984; Efron & Tibshirani, 1991; Venables & Ripley, 1997). These homogenous groups are called terminal nodes and contain the mean value of all the data points in each terminal node (Yang et al., 2017). As a result, the tree like series of tests can be used to predict a likely value of the dependent variable for new data (Lawrence & Wright, 2001). The CART algorithm can be seen as a greedy search algorithm as the splits that are being made at every step are only based on the best current option, without considering the splitting performance of upcoming tests (Kotsiantis, 2013).

When CART generates large trees, it often overfits the training data, resulting in poorer generalization performance to unseen samples (Yang et al., 2017). To prevent this, the tree can be made smaller with pruning to be more robust (Lawrence & Wright, 2001). Pruning is often referred to as post-pruning and is considered a backwards selection method (Prodromidis and Stolfo, 1998; Prodromidi & Stolfo, 2001). It sequentially removes the splits and prunes the tree from its maximum size back to the root node, creating a set of consecutive nested candidate trees of decreasing size (Yang et al., 2017). Each candidate tree is built on the training data and the tree with the lowest prediction error is selected as the final tree (Breiman, 2001). Pre-pruning on the other hand is conducted during recursive binary partitioning and aims to prevent splits that do not meet certain criteria such as a minimum number of observations for a split to happen or a minimum number of observations in a (terminal) node itself (Kotsiantis, 2013). While post-pruning is very simple and interpretable, using average values at the terminal nodes for prediction often results in compromised prediction performance (Antipov & Pokryshevskaya, 2012; Bayam et al., 2005; Bel et al., 2009; Loh, 2011).

There are several different pruning methods. Cost complexity pruning (CCP) is a pruning method that creates a several subtrees from the original large tree by progressively pruning the tree to its root node (Prodromidis & Stolfo, 2001). Prodromidis & Stolfo explained that CCP does this by first associating a complexity measure $C(T)$ based on the number of terminal nodes (T) of the fully grown decision tree. It then prunes the tree by minimizing a cost complexity metric $R\alpha(T)$. $R\alpha(T)$ is defined as:

$$R\alpha(T) = R(T) + \alpha * C(T)$$

where $R(T)$ represents the misclassification cost or the error rate of the tree and α is a complexity parameter which is larger than zero (Prodromidis & Stolfo, 2001). The authors also explained that the strength of pruning of the initial tree is controlled by α. When α decreases, the final size of the tree increases as the penalty for having many terminal nodes decreases. A search algorithm can be used to calculate every α value that changes the size of the tree (Bradford et al., 1998). Bradford et al. explained that the parameter can be chosen by using cross-validation.

According to Henrard et al. (2015) DTs have many key benefits compared to more conventional methods. The authors explained that they are easy to understand and interpret, as they allow individuals to easily assess which subgroup a specific observation belongs to. Moreover, the authors noted that DTs can handle nonlinear relations and make no assumptions about the distribution of variables, unlike OLS, logistic regression, and LASSO models. Furthermore, they highlighted that DTs can easily handle multicollinearity by selecting the best splits at each node. Additionally, they explained that DTs can identify outliers, because outliers can easily be separated by separate nodes. Finally, they stated that DTs can handle interactions between explanatory variables by showing them in the tree with the best split at each node. Despite all the advantages, Friedman & Meulman (2003) emphasized that the major disadvantage of DTs is that they tend to be less accurate. While DTs can sometimes be competitive, they are never the most accurate model in any given application (Friedman & Meulman, 2003).

2.6 Random Forest

According to Breiman (2001), many of the disadvantages of decision trees have been dealt with by growing an ensemble of trees instead of using only one decision tree. One of these techniques is an RF, which was proposed by Breiman (2001) himself. An RF is a tree ensemble method that uses multiple CART trees for prediction (Breiman, 2001). RF can be used for classification or regression problems and is gaining a lot of popularity in several research fields because of its inherent non-linear and non-parametric characteristics (Desai & Ouarda, 2021).

When an RF is used for regression tasks, it is called random forest regression (RFR) (Desai & Ouarda, 2021). Desai & Ouarda explained that RFR first creates multiple samples from a given set of training data by using bootstrapping. Bootstrapping is a resampling method, developed by Efron (1979), that resamples observed data with replacement (Akins et al., 2005). Akins et al. (2005) explained that each bootstrap sample is a random subset drawn with replacement from the original data, with the same size as the original data set. Breiman (2001) explained that after the samples are created, RFR grows full tree-structured classifiers $h(x, \theta_k)$ on these samples where x represents the independent input variables, θ_k represent independent identically distributed random vectors and k identifies each individual tree, ranging from 1 to K (Breiman, 2001). θ_k take on numerical values in RFR instead of categorical values, which would be the case for a classification task (Desai & Ouarda, 2021). Finally, the results of all fully grown trees are combined, and an estimate of the target variable is obtained by taking the average over the individual trees (Larivière & Van den Poel, 2005). When RF would be used for a classification task, the most popular class from the predictions of the ensemble of trees would be selected as the final predicted value (Breiman, 2001).

RFR uses a random selection of a subset of all the original independent variables to grow every individual tree (Larivière & Van den Poel, 2005). This random subset is used to split the data into homogenous groups and is much smaller compared the original number of independent variables that were selected for the analysis. According to Breiman (2001), this number should be equal to square root of the original total number of independent variables. He explained that the variable randomness minimizes correlation while still maintaining strength. He found that by doing this, the model has an accuracy that compares favorably with a Adaboost, which is a different tree ensemble method.

Because RFR uses bootstrapping, one third of the observations from the original data set are left out in the bootstrap samples, which are known as out-of-bag (OOB) Samples (Desai & Ouarda, 2021). The estimated error on the OOB samples is called the OOB-error rate, which can be used as an accuracy measure itself instead of validating RFR on a test set and to select the optimal number of trees used by the model (Desai & Ouarda, 2021). A captivating by-product of RFR is the variable importance, which compares each independent variable on how useful they are for predicting the dependent variable (Ishwaran et al., 2004).

In contrast to conventional methods, RFR has several nice advantages such as its non-parametric nature, its strong prediction accuracy, and its ability to calculate the variable importance (Ouedraogo et al., 2018). Moreover, Desai & Ouarda (2021) explained that RFR handles noise and outliers quite well, as the input training sets are randomly sampled by replacement, and because the subset of variables used by each tree are selected randomly. Moreover, they stated that the absence of correlation between individual trees prevents RFRs from overfitting the training data. Additionally, Larivière & Van den Poel (2005) suggested that RFR needs little computation time and is easy to use, because the only two parameters that need to be determined are the number of trees used by the model and the number of estimators that need to be randomly selected from the original predictors (Larivière & Van den Poel, 2005). Breiman (2001) recommended to use many trees and to use the square root of the number of predictors used for the number of estimators used by each tree. Despite all its advantages, a big disadvantage of the RF is its black-box nature, making it hard to interpret the complex relationships between the dependent variable and the predictors (Zhang et al., 2021).

2.7 Feature selection methods

Feature selection reduces the number of input variables during predictive model development (Brownlee, 2019). Brownlee explained that it is often desirable to minimize the number of input variables as this decreases the computational cost of modeling and can improve the model performance as well. According to Brownlee, there are two types of feature selections methods, namely supervised feature selection methods and unsupervised feature selection methods. Supervised feature selection methods use the target variable as a basis for selecting the most relevant variables. Unsupervised feature selection methods, on the other hand, do not use the target variable and are based solely on the characteristics of the features themselves. Supervised feature selection methods can be divided into filter methods, wrapper methods, embedded methods, and hybrid methods (Jović et al., 2016).

According to Jović et al. (2016) filter methods select, and rank features based on a performance measure. They explained that only after the best features are found, the modeling algorithms can use them (Jović et al., 2016). According to the authors, filter methods can either rank individual features or evaluate entire feature subsets. They explained that measures for feature filtering include information, distance, consistency, similarity, and statistical measures. Furthermore, they stated that filter methods are classified by classification, regression, or clustering tasks.

Wrapper methods use a specific learning algorithm to assess the feature subsets based on classification error estimates and to construct the final classifier (Chen & Jeong, 2007). Wrappers are slower than filters because they depend on the algorithm's resource demands (Jović et al., 2016). Jović et al. (2016) also explained that the feature subsets generated by wrappers are biased towards the algorithm used for evaluation. Therefore, the independent validation sample and another algorithm are used after the final subset is found for a reliable general error estimate (Jović et al., 2016). However, the authors explained wrappers perform better than filters as they evaluate subsets using a real modeling algorithm.

Jović et al. (2016) explained that embedded methods select features during the algorithm's execution and that these methods are thus embedded in a model algorithm itself. Moreover, they explained that some embedded methods perform feature weighting based on regularization models with objective functions that minimize fitting errors and, in the meantime, force the feature coefficients to be small or to be exact zero. They stated that these methods based on LASSO and Elastic Net usually work with linear classifiers and induce penalties on irrelevant features.

Hybrid methods combine filters and wrappers (Jović et al., 2016). Jović et al. explained that these methods first use a filter to reduce feature space, which generates candidate subsets. After that, a wrapper is used to select the best candidate subset (Jović et al., 2016). Moreover, the authors explained that hybrid methods often achieve accuracy and efficiency, and various filter-wrapper combinations can be used for hybrid methodology.

2.8 Best subset selection

BSS tries to find a small subset of estimators, that results in the best prediction accuracy when used by a certain model (Hocking & Leslie, 1967). Given a dependent variable $Y \in R^n$ and a predictor matrix $X \in R^{n*p}$ containing estimators and a subset with a size of k predictors, where k ranges between 0 and $min\{n, p\}$ (the minimum value between the number of observations (n) and the number of predictors (p)) and where R represents the set of real numbers, BSS tries to find the best subset of k predictors that give the best fit in terms of the squared error (Hastie et al., 2020). BSS does this by minimizing the squared difference between the observed values and the predicted values of the dependent variable, subject to a condition where the number of nonzero coefficients in β is smaller than or equal to k.

$$\min_{\beta \in R^p} ||Y - X\beta||_2^2 \;\; subject\ to\ ||\beta||_0 \leq k, \beta \in R^p,$$

where, $||\beta||_0 = \sum_{i=1}^{p} 1\{\beta_1 \neq 0\}$ is the l_0 norm of β (Bertsimas et al.,2016). Bertsimas et al. also explained that l_0 counts the number of nonzero elements in the vector β and $1(\cdot)$ denotes the indicator function. Moreover, they explained that the condition $||\beta||0 \leq k$ ensures that the number of estimators used by the model is limited, which helps to prevent the model from overfitting the training data and keeps it simple. Moreover, the indicator function $1(\cdot)$ takes a value of 1 when the condition inside it is true and 0 when it is false.

One problem with BSS is that several BSS algorithms, like leaps in R, do not scale to problem sizes where p > 30 (Bertsimas et al., 2016). Due to this limitation, best subset selection is often considered as impractical for problem sizes where the number of predictors is larger than 30, leading to the method being dismissed by the statistical community (Bertsimas et al., 2016).

2.9 Recursive feature elimination

RFE is a feature selection method developed for small sample classification problems (Guyon et al., 2002). However, RFE is also used for regression problems (Ai, 2022; Qiu et al., 2011; Zhou et al., 2009). RFE attempts to select the optimal feature subset based on the learned model and the classification accuracy (Jeon & Oh, 2020). Traditional RFE sequentially removes the worst feature that lowers the model accuracy after building a model (Chen et al., 2007; Jeon & Oh, 2020). However, according to Jeon & Oh (2020), the new RFE approach tries to improve the model performance by removing the worst features based on feature importance instead of classification accuracy, which is based on the support vector machine (SVM) model. Jeon & Oh called this method feature-importance-based RFE and stated that it involves training a classifier with a dataset and obtaining feature weights to determine importance. They explained that the feature with the lowest weight is removed, and the classifier is re-trained until no features remain. Moreover, they stated that with this approach, RFE can also be applied to other classification models such as RF models and gradient boosting machines (GBMs), which both have a built-in feature evaluation mechanism. According to Jeon & Oh (2020), RFE is an embedded selection method, while Ai (2022) and Jović et al. (2016) both stated that it is a wrapper method.

Chen et al. (2007) stated that RFE works well in small-sample feature selection, but often removes useful redundant features and weak features in bigger data sets. The authors found that redundant features can improve class separation and that using weak features together can significantly improve performance. Moreover, they found that by removing these features, classification performance could be reduced. Darst et al. (2018) supported this in their own research by integrating RFE with a RF model, calling it a Random Forest-Recursive Feature Elimination algorithm (RF-RFE). Darst et al. (2018) explained that correlated predictors weaken the ability of a RF to identify strong predictors and that RF-RFE can solve this issue in small datasets but that its effectiveness in high-dimensional datasets was still unknown. Furthermore, the authors stated that RF-RFE reduced the importance of correlated variables when using high dimensional data, but that it also decreased the importance of causal variables in the presence of many correlated variables. They explained that this made it difficult to detect both, suggesting that RF-RFE may not be suitable for high-dimensional data.

Furthermore, Ai (2022) integrated RFE with a LASSO model calling it a LASSO-Recursive Feature Elimination (LASSO-RFE) algorithm. He explained that LASSO-RFE utilizes LASSO as the underlying algorithm and enhances the feature selection using RFE, keeping the benefit of LASSO in effectively removing irrelevant variables while improving model interpretation, which makes the selected features more representative. Moreover, Tyagi et al. (2021) combined RFE with a DT calling it a Decision Tree-Recursive Feature Elimination (DT-RFE). They explained that DT-RFE removes the least important variables in a recursive manner, based on the feature importance obtained from the DT.

2.10 Evaluation metrics

Evaluation metrics are numerical indicators utilized to evaluate the performance and effectiveness of statistical or machine learning models (Saxena et al. 2008). As a result, these metrics make it possible to compare different models in terms of performance to identify the best performing model. When evaluating model performance, it is important to choose the right evaluation metric(s) (Tapper, 2022). According to Botchkarev (2018), the mean square error (MSE), the root mean square error (RMSE), the mean absolute error (MAE) and the mean absolute percentage error (MAPE) are the most popular metrics used for regression models in the past decades.

Chicco et al. (2021) stated that the MAE and the MSE are the two basic members of the family of metrics that evaluate model performance by looking at the distance of the predicted values to the actual values.

The MAE is defined as the mean of the absolute difference between the predicted and actual values while the MSE is defined as the mean of the squared difference between the predicted and actual values (Chai & Draxler, 2014). Moreover, MAPE is a derivation of MAE. MAPE is defined as the average percentage errors between the predicted values and the actual values (De Myttenaere et al., 2016). Similarly, RMSE is a derivation of MSE. RMSE standardizes the units of measures of MSE and is defined as the square root of the mean of the squared difference between the predicted and actual values (Chai & Draxler, 2014; Chicco et al., 2021). Another commonly used evaluation metric is the coefficient of determination, which is better known as R-squared and is proposed by Wright (1921). It explains by how much the dependent variable is explained by the independent variables, in terms of variance (Chicco et al., 2021). However, it does not imply causality (Tapper, 2022). R-squared is upper bounded by the value 1. If it is equal to 1, the variance of the target variable is fully explained by the independent variables, and when it is equal to 0, these variables do not explain any variance (Chicco et al., 2021). In contrast, the values of MAE, MSE, RMSE and MAPE cover the entire positive branch. For these metrics, zero implies a perfect fit, while larger values imply poorer model performance (Chicco et al., 2021).

In studies where similar models are being used for regression tasks, R-squared, RMSE and MAPE seem to be quite popular metrics. For instance, Gomes et al. (2020) and Gomes & Jelihoyschi (2020) utilized R-squared to evaluate the performance of an RT model. Similarly, Ouedraogo et al. (2018) used the RMSE to evaluate an RFR model, while Thach et al. (2021) used RMSE and MAPE to evaluate a LASSO regression model. Additionally, in a related line of research where CLV is predicted with machine learning R-squared, MSE, RMSE and MAE seem to be popular. For example, Venkatakrishna et al. (2021) evaluated their models using R-squared, and MSE, while Tapper (2022) used the MAE, RMSE and R-squared.

Due to the errors being squared, MSE and RMSE are relatively sensitive to outliers (Chai & Draxler, 2014; Chicco et al., 2021). Willmott & Matsuura (2005) stated that evaluation metrics based on the sum of squared errors have "disturbing characteristics" and should not be used as evaluation metrics. Both Chai & Draxler (2014) and Willmott & Matsuura (2005) strongly advised to use MAE instead of MSE or RMSE. According to Tapper (2022), the advantages of MAE are that it is very simple and interpretable. Furthermore, MAPE has a strong bias towards underestimating large forecasts, making it an inappropriate evaluation metric when large errors are anticipated (De Myttenaere et al., 2015). In addition, Foss et al. (2003) concluded that MAPE is "unreliable and may have misled the entire software engineering discipline". Moreover, Chicco et al. (2021) concluded that R-squared is more informative and interpretable compared to MSE, RMSE, MAE and MAPE. They suggested that R-squared should be used as a standard evaluation metric for regression analyses. However, Li (2017) stated that the R-squared should not be used as an evaluation metric for models when numerical data is used, because R-squared is biased and misleading when predicted and observed values are not perfectly matched.

3 Data

3.1 Raw data

The data used in this research is user subscription data from KKBOX, containing information about, demographics, transactions, and churn related features from subscribers. The data was created for a churn prediction competition on Kaggle but was used to predict CLV in this research. Kaggle divided the KKBOX data into five separate data sets, but only three of them were used in this research. This is because the third and fourth data set originally served for training and test purposes respectively. However, in this research, the third data set was only used to extract a variable. As a result, the very similar fourth data was obsolete for this research. Moreover, the fifth data set was unmanageable[1] in R. The three raw data sets that will be used are discussed below.

The first raw data set is called transactions.csv. It contains a collection of records capturing financial interactions and contains the following variables: *msno, payment_method_id, payment_plan_days, plan_list_price, actual_amount_paid, is_auto_renew, transaction_date, membership_expire_date and is_cancel*. The variable descriptions can be found in table A1 in the appendix. Furthermore, the data contains 21,547,746 transactions, made between 2015-01-01 and 2017-02-28.

The second data set is called members.csv and mainly contains user demographics and information about their subscription and contains the following variables: *msno, city, bd, gender, registered_via, registration_init_time*. The variable descriptions can be found in table A1. Moreover, the data set contains 6,769,473 users. However, one variable in the data set seems to be missing[2].

The third data set is called train.csv and contains only two variables, namely *msno* and *is_churn*. The variable descriptions can also be found in table A1. Additionally, the data set contains 992,931 users.

3.2 Data sampling

The three previously mentioned data sets are extremely large data sets. To reduce computation time, a random sample of 100,000 user will be used instead of the entire userbase. However, the three data sets do not contain the exact same users. Because train.csv contained the least number of users, the 100,000 users were randomly selected from the train.csv data set, and only the 100,000 selected users were kept in the individual three data sets, resulting in 1,600,435 observations in the sampled transaction data, 88,472 users in the sampled members data and 100,000 observations in the sampled train data. Members.csv ended up with less than 100,000 users, because the train.csv data contained some users that were not present in the members.csv data. However, this ‘issue’ was solved later when the data sets were merged, which is explained in the final paragraph of section 3.3.

[1] The unmanageable data set is unmanageable due to its large size (30GB). Despite using various R packages such as `data.table` and `vroom`, it could not be loaded into R.

[2] The variable named *expiration_date* seems to be missing in the members.csv data, which should be present according to Kaggle.

3.3 Data cleaning

After sampling, the data sets needed to be cleaned. In the sampled members data set, the variable *bd*, which represents the age of the customer at the time that the data was collected, contained some odd values above 99, even reaching 1032. Additionally, *bd* contained values equal to negative values or zero. Furthermore, the variable *gender* frequently exhibited a blank value and the respective bd value was often equal to zero for these users. Therefore, it is likely that customer demographic information is missing for these users. To improve the analysis, all the suspicious users were removed from the data.

Besides the sampled members data set, the sampled transactions data set also needed to be cleaned. For some of the observations in these data set, *actual_amount_paid* was equal to zero, which clearly does not make any sense. Moreover, some transactions seemed to be duplicated as some users had supposedly made multiple transactions on the same exact day with the same exact price, which seems odd for a subscription service. As a result, these 'duplicated' observations and observations where *actual_amount_paid* was equal to zero were removed from the sampled transactions data set.

In between the cleaning process, the cleaned sampled members data and the sampled train data were merged, creating the data set called user data. The final step of the cleaning process was to remove the deleted user IDs of the sampled members data from the sampled transactions data and to remove the (possibly) deleted user IDs of the sampled transactions data from the newly created user data.

3.4 Transforming variables

After cleaning the data, some variables still had to be transformed to make sure that additional preprocessing steps worked properly. First, all the integer date variables (with values such as "20150202") were transformed into proper date variables with a %Y%m%d format. Secondly, variables with character values were changed into factor variables because Barrowman (2020) stated that although most statistical operations in R will transform character variables automatically into factors first, it is more efficient to convert them manually. Moreover, every integer binary variable, *city*, *registered_via* and *payment_method_id* (see table A1 in the appendix)) were also changed to factors, because these variables were categorical in nature. Integer variables that are categorical in nature should be converted to factors to prevent the ordered numeric values to take on a meaning (Bhalla, 2016).

3.5 Aggregating the data, removing variables and creating new variables

The next step of the data preprocessing was to aggregate the sampled transactions data in a way that each unique user ID appeared only once in the aggregated transactional data. For this aggregated transaction data, *first_transaction* (date), *last_transaction* (date), *frequency*, *total_amount_paid* and *average_amount_paid* were created/calculated for each user. With these new variables, *recency_months* (the recency in terms of months instead of days), *number_months_from_start* (the number of months between the start of the period and the first purchase) could also be created. The final variable that needed to be created was *CLV*. This step is explained in detail in the methodology section.

Moreover, in the data aggregating step, some uninformative variables and time dependent variables that varied over time were left out from the new aggregated transactional data. These variables were, *payment_method_id*, *payment_plan_days*, *plan_list_price*, *is_auto_renew*, *membership_expire_date* and *is_cancel*. Furthermore, after creating all the variables for the analysis, the variables (*first_transaction, last_transaction* and *total_amount_paid*) that only served the purpose to create new variables were removed as well. Finally, the name of *msno* was changed to *user_ID.*

3.6 Merging the data

One of the final steps of the data preprocessing was to merge the newly created aggregated transactional data set with the user data set created a few steps back, resulting in the final aggregated data set. This data set included 11 variables (*user_ID*, *gender*, *bd*, *city*, *registered_via*, *is_churn*, *frequency*, *average_amount_paid*, *recency_months*, *number_months_from_start* and *CLV*) and consisted out of 38,274 observations. The variable descriptions of the full aggregated data can be found in table A2 in the appendix.

3.7 Descriptive statistics

In table 1, the descriptive statistics can be seen for all the numerical variables from the final aggregated data set.

Table 1 Descriptive statistics of the numerical variables used for analysis

Variable	Mean	Median	Standard deviation	Min	Max
bd	30.01	28.00	8.90	1.00	98.00
frequency	17.31	19.00	8.16	1.00	42.00
average_amount_paid	187.17	149.00	195.52	35.00	2000.00
recency_months	0.61	0.00	1.91	0.00	22.00
number_months_from_start	5.63	2.00	7.53	0.00	25.00
CLV	8324.31	8865.82	3483.82	156.58	18376.77

Besides numerical variables, this research also used binary factor variables for the analysis. In table 2, the descriptive statistics of the binary factor variables can be seen.

Table 2 Descriptive statistics of the factor variables used for analysis

Variable	Level	Count	Proportion
gender	male	20222	0.53
is_churn	1	3311	0.09

Note: The first level of the binary factor variable are left out of this table.

3.8 Bar plots of the remaining factor variables

In figure 1, the frequencies of the different levels of the factor variables *city* and *registered_via* can be seen in two bar plots next to each other. What is important to note is that the reference category (the first level of the factor variable) is still present for both variables.

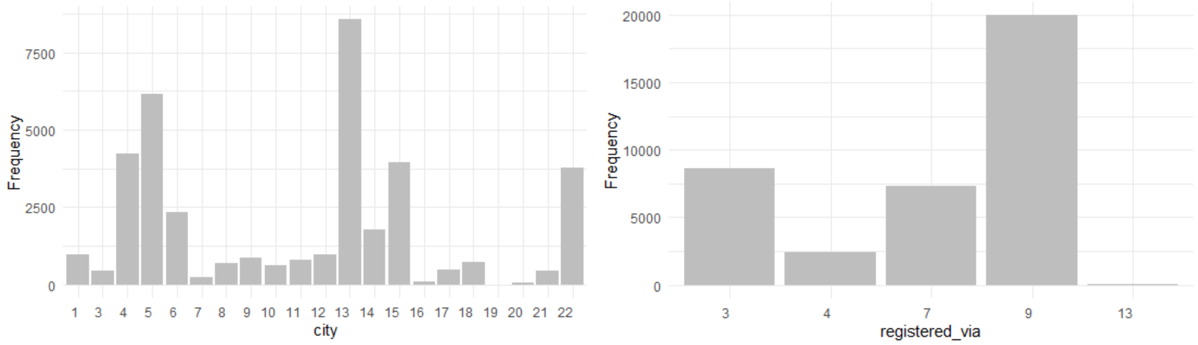

Figure 1 Distributions of the numerical variables used for analysis

3.8 Data distributions

In figure 2, the distributions of the numerical variables can be seen. What is important to note is that this figure contains multiple histograms, each using a different scale.

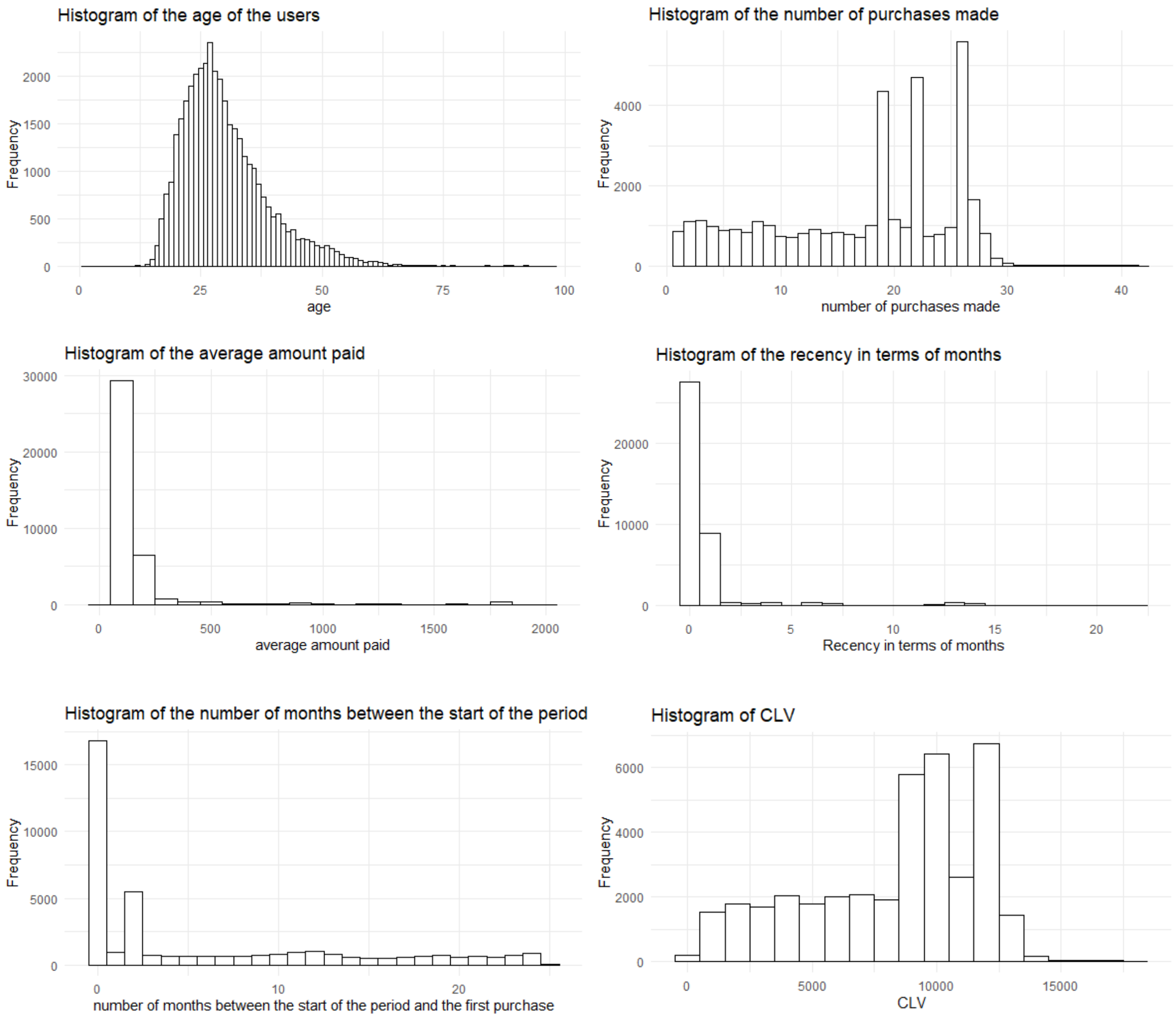

Figure 2 Distributions of the numerical variables used for analysis

3.9 Data correlations

In figure 3, the correlations between all the numerical variables are visualized in a correlation plot. A large dark blue circle indicates a positive correlation close to "1", while a large dark red circle indicates a negative correlation close to "-1". The smaller and lighter the circle becomes, the weaker the correlation between the two variables. Figure 9 shows a few very intuitive high correlations between variables, such as the positive correlation between *frequency* and *CLV*, the negative correlation between *number_months_from_start* and *CLV*, and the negative correlation between *frequency* and *number_months_from_start*. The latter suggests the possibility of multicollinearity between these two variables. Moreover, it is interesting to observe a very low negative correlation between *average_amount_paid* and *CLV*, and a high positive correlation between *average_amount_paid* and *recency_months* which also seems to suggest multicollinearity.

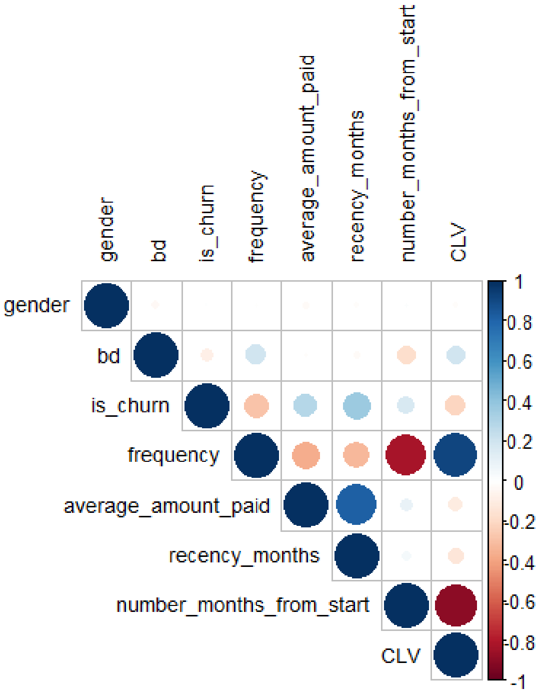

Figure 3 Correlation plot of all the numerical variables used for analysis

3.11 Dummy coding

The final data preprocessing step was to transform all factor variables into dummies with dummy coding. For each variable, the reference category was removed to prevent multicollinearity. For example, when a factor variable with five levels is transformed with dummy coding, it is transformed into four dummies. As a result, the final data set used for analysis contained 31 variables instead of the 11 variables mentioned in section 3.6.

3.12 Creating training data and test data

After the data preprocessing was completed, the data had to be split up in training data and test data to predict CLV. According to Radečić (2022), a time-based split is very common way to split time dependent data in training and test data for time dependent data. He explained that when predicting data with time series data, the values at the rear of the data set should be used for testing and everything else for training. However, I believe that this is not the case for this research, as supervised learning will be used for prediction instead of a prediction method that forecasts future values of the dependent variable, without needing future values from the predictor variables. Moreover, the final data used for prediction is an aggregated transactional data set instead of a time series. Additionally, when the original transaction data (explained in section 3.1 to 3.4) would be split up based on a threshold date, it would be more difficult to create a CLV variable for the training data and test data because these data sets would not cover the same number of months. As a result, I decided to use a stratified split to randomly split users into a training data set and a test data set.

According to Gholamy et al. (2018), empirical research has shown that allocating 20-30% of the data for testing and 70-80% for training results in the best accuracy. In this research, 70% of the data was randomly allocated for training purposes, and 30% for testing purposes. A 70/30 split was used instead of an 80/20 split due to the substantial size of the dataset. Consequently, I was confident that allocating 70% of the data for training would be sufficient, allowing for a larger portion of data to be reserved for validation.

4 Methodology

The methodology explains how the data analysis is conducted in this research. Section 4.1 explains how CLV was calculated. Moreover, section 4.2 to section 4.10 explain how a LASSO regression, LASSO-BSS, LASSO-RFE, RT, RT-BSS, RT-RFE, RF-BSS and RF-RFE were created and tuned in R to predict CLV. Finally, section 4.11 explains how the prediction accuracy of the nine supervised learning models were measured.

The reason for including several LASSO regression models was because these models are quite easy to interpret. Therefore, it seemed interesting to compare the prediction performance of these models with black-box supervised machine learning models like the RF models, which often have high prediction accuracy, but are hard to interpret. By including both models, a trade-off can be made between model interpretability and model accuracy. Furthermore, several RT models are also included. The reason for including these RT models is that it seemed interesting to see by how much prediction accuracy increases when a model uses multiple trees instead of just one.

Furthermore, the reason for including best subset selection as the first additional feature selection method was because it is well established method with many successors. Therefore, it seemed interesting to see how well this well-established feature selection method would affect the prediction performance of the supervised learning Moreover, RFE was included as a second feature selection method, to assess the extent to which a newer feature selection method could improve accuracy in contrast to the well-established method.

4.1 Calculating CLV

Section 2.1 and 2.2.1 from the literature review explained that CLV can be calculated with the net present value of the profit gained from a user. However, the KKBOX data used in this research only included information about revenues gained from users and not about incremental costs. Therefore, instead of using the profit gained from a user, the revenue gained from a user was used to calculate CLV. As a result, the CLV in this research was calculated in the following way:

$$CLV = E\sum_{t=0}^{T}\frac{R_t * r^t}{(1+d)^t}$$

where R_t represents the revenue earned from a user during period t, T represents the customer lifetime, r represents the retention rate and d represents the discount rate used to discount cash flows of future periods. t starts at zero, which represents the current period, and each period is equal to 26 months. Furthermore, no discount rate was given in the data. As a result, the discount rate was based on commonly used discount rates. According to Blattberg et al. (2009), the annual discount rate often ranges between 10% and 20%. Based on this information, this research used an annual discount rate of 10%. However, in this research, one period is equal to 26 months. As a result, the annual discount rate needed to be transformed into a discount rate for 26 months. This transformation involved using a simple annual discount rate instead of a compound discount rate, as it was assumed that compounding CLV is unnecessary. A simple annual discount rate of 10% translates to a 21.67% discount rate over 26 months.

The next step was to determine r, which can be defined as $r = \frac{E-N}{S}$, where E represents the number of customers at the end of a specific period, N represents the number of new customers acquired during the period and S represents the number of customers at the beginning of the period (Blattberg et al., 2009). However, calculating an accurate retention rate was quite hard with the given data. Kaggle seemed to suggest that the data set has a contractual setting. However, it seemed to be common for users to unsubscribe and resubscribe once or multiple times in a short time. As a result, it was not clear whether a user churned or not. As a result, when using $r = \frac{E-N}{S}$ to calculate the retention rate, E could have been underestimated. When only the last month of the data was considered for E, the retention rate was equal to approximately 0.66. However, when the two last months or the three last months were also considered for E, the retention rates are approximately 0.97 and 0.98 respectively. These large differences seemed to suggest that only taking the last month of the period into account would underestimate E. As a result, users that were active in either one of the last two months or in both, represented E.

The final step to calculate CLV was to determine T. Bonacchi & Perego (2012) explained that customer lifetime represents the entire period during which a person is a customer of a certain company without churning. Moreover, they stated that the average lifetime is determined by the rate at which customers cancel their subscriptions during a period. In other words, T can be defined as $T = \frac{1}{1-r}$ where r represents the retention rate. However, when using the retention rate from the previous paragraph (0.97), another problem arises. The formula to calculate T would suggest that T is equal to approximately 34.01 periods, which is equal to approximately 74 years. However, it seemed unlikely that users subscribe to the same music streaming service for almost 74 years, due to changes in the market during such a long period. However, Malthouse & Blattberg (2005) suggested that CLV should be estimated over a long period of time rather than a customer's entire lifetime. They explained that hotels and airlines evaluate CLV for only one year. However, one year seemed too short for the music streaming industry. Therefore, T was set to "5", which is equal to 13 years. What is important to note is that the current period ($t = 0$) was also included for the CLV calculation. As a result, the current period and the following five future periods were used to estimate CLV.

The reason for choosing the basic CLV calculation method to estimate CLV was because Kaggle seemed to suggest that there was a contractual setting. With a contractual setting, it is often very easy to predict future revenues from customers, as they pay a monthly or yearly fee. This fee often stays constant unless the contract is changed. Moreover, with a contractual setting, users keep paying their fees until they churn. As a result, using historical revenue data together with a discount rate and a retention rate seemed to be a sufficient method to estimate CLV in a contractual setting.

4.2 LASSO regression

The first model that was used to predict CLV is a LASSO regression. The first step was to set a seed for reproduction purposes. After that, cv.glmnet() from the `caret` package in R (Kuhn, 2023) was used to create the model, in which "nfolds" was set to "10" and "type.measure" was set to "MSE". Moreover, cv.glmnet() needs a matrix containing the predictor variables and a numeric string representing the dependent variable as input data to be trained. The matrix, created with as.matrix(), and the numeric string were both obtained from the training data. After the LASSO regression was created with the cv.glmnet() function, the coefficient paths of the predictors were plotted, which showed how the coefficients of the predictors react to different values of lambda. This plot was created as it provides some insights into the variable importance of the predictors used by the LASSO regression model.

The next step was to obtain the optimal value of lambda, which is the lowest value of lambda. This value was used in the glmnet() code to create the final lasso model. With the coef() function, the coefficients of the predictor variables were obtained. This function also showed which coefficients were shrunk to zero, making it clear which variables were used for prediction. The final step was to predict the CLV values from the test data with the predict() function, which used the final lasso model with the optimal lambda.

4.3 LASSO-BSS

The second model that was be used to predict CLV is a LASSO-BSS regression. The first step was to perform BSS with a linear regression using the regsubsets() function from the `leaps` package (Lumley, 2020). This function uses the entire training data as input, and "nvmax" was set to "31" so that the maximum size of the subsets was equal to the number of predictors. The reason for using a linear regression as the initial model for BSS was because to my knowledge, an R package performing BSS while using LASSO as the initial model does not exist. As a result, a linear regression was selected as it is the most similar supervised learning algorithm that was available compared to the linear LASSO regression.

The next step was to find the best subset. With the summary() function, several statistics can be obtained from the regsubsets() function to select the optimal subset, such as the adjusted R-squared values, Mallows' Cp values, and Bayesian Information Criterion (BIC) values for all the possible subsets. The adjusted R-squared measures the goodness of fit of the model, while penalizing the addition of unnecessary predictors that might cause overfitting. When using the adjusted R-squared, higher values are preferred. Moreover, the Cp statistic measures the trade-off between the goodness of fit and the model complexity. Furthermore, The BIC, balances the goodness of fit and the model complexity, but uses a Bayesian approach. When using the Cp statistic or the BIC, smaller values are preferred. This research used the adjusted R-squared to select the best subset, because the adjusted R-squared seemed to exhibit less inclination towards simpler models compared to the Cp statistic or the BIC. This seemed favorable, as the next step would fit a LASSO regression on the best subset, which shrinks irrelevant variables towards zero, if they would still be present in the best subset.

After the best subset of variables was found, the steps explained in section 4.2.1 for the LASSO regression were repeated. The only difference with those previously explained steps was that only the variables included in the best subset were used, instead of using all predictors.

4.4 LASSO-RFE

The third model that was used to predict CLV is a LASSO-RFE model. The first step was to create another LASSO regression model (after setting a seed) using the same first few steps as in section 4.2.1, but this time, using a standardized data set. The reason for this is that RFE needs the variable importance order to work properly. One way to obtain the variable importance order for the LASSO-RFE model is to look at the coefficients of the LASSO regression when it is performed on standardized data. When the data is standardized with the scale() function in R, the data is transformed by setting the mean of each variable to "0" and the standard deviation to "1". This allows a fair comparison and analysis of variables with different scales. As a result, the absolute values of the coefficients represent the variable importance of the variables used by the LASSO-RFE model.

After the variable importance order was obtained, the matrix including the predictor variables used in section 4.2.1 was transformed in a way that the order of the columns was like the obtained variable importance order. This transformed matrix and the numeric string containing the CLV values, were then used as the input data for RFE (after setting a seed) with a linear regression as the initial model. A linear regression was used as the initial model instead of a LASSO regression because I could not find a package in R that performed RFE with a LASSO regression as the initial model.

In order to set some specifications for RFE, the rfeControl() function from the `caret` package (Kuhn, 2023) was used, where "functions" was set to "lmFuncs", "method" was set to "cv" and "number" (number of folds for the cross-validation) was set to "10". After that, the rfe() function from the `caret` package (Kuhn, 2023) was used to create the RFE model using linear regression as the initial model, using the specifications from the previous rfeControl() function. Moreover, "sizes" was set to "1:31", so that the maximum size of the subsets was equal to the number of predictors. After performing RFE, the optimal features were obtained by using the $optVariables in R.

After obtaining the optimal features from RFE, the steps explained in section 4.2.1 for the LASSO regression were repeated. The only difference with those previously explained steps was that only the optimal variables were included in the input matrix, instead of using all predictors.

4.5 Regression tree

The fourth model that was used to predict CLV is an RT model based on the CART learning algorithm introduced by Breiman et al. (1984). The rpart() function from the `rpart` package (Therneau et al., 2022) was used to train the RT using the normal training data as input data. Moreover, the complexity parameter (cp) was set to "0" for the control parameter, so that the algorithm would build a full tree. Furthermore, "method" was set to "anova", so that it was clear for the algorithm that the tree was used for a regression problem.

In the literature review, it was explained that CART analyzes all possible 'splits' for every explanatory variable and then selects the split, which reduces the deviance in the dependent variable the most (Breiman et al., 1984; Efron & Tibshirani, 1991; Venables & Ripley, 1997). However, when using the anova method from the rpart() function, the lowest Sum of squares total (SST) of the node minus the sum of squares left (SSL) of the left 'son' and the sum of squares right (SSR) of the right 'son' (SST − (SSL + SSR)) will be used for the splitting criteria, where SST is equal to $\sum(y_i - \overline{y})^2$, y_i represents the individual observed values in a data set and $\overline{y}$ represents the mean (average) of the observed values in a data set (Therneau et al., 2022).

After the full RT was created in R, the model was plotted with the plot() function for visualization. However, because of the large size of the tree, information about the splits and the nodes were excluded to prevent a very messy plot. The next step was to create a cp table with $cptable to find the cp value that resulted in the lowest xerror (cross-validated error). This cp value was needed to optimally prune the tree with the CCP method, explained in section 2.5. A higher cp value results in a smaller tree.

With the optimal cp value, the final pruned tree could be created. This was done by using the same rpart() code that was used to create the full grown tree, except for the cp value, which was set to the optimal value instead of zero. This final model was then used to predict the CLV values in the test set using the predict() function.

After the CLV values were predicted, a new pruned tree was created with a relatively high cp value and plotted with the rpart.plot() function from the `rpart.plot` package in R (Milborrow, 2022), showing the first few tests and splits of the RT model. This additional step was needed because the full-grown tree and the optimal pruned tree were too large to include information about the tests and the nodes.

4.6 RT-BSS

The fifth model that was used to predict CLV is an RT-BSS model. However, when I tried to create this model in R, the same problem that was discussed in section 4.2.2 arose as I could not find a best subset selection algorithm in R that uses a regression tree as the initial model. As a result, BSS again used a linear regression as the initial model, and the best subset found in the LASSO-BSS model was used again for the RT-BSS model. With these selected variables instead of using all the predictor variables, the steps explained in section 4.2.4 were repeated.

4.7 RT-RFE

The sixth model that was used to predict CLV is an RT-RFE model. However, when I tried to create this model in R, the same problem that was discussed in section 4.2.3 arose, as I could not find a package in R that performed RFE with an RT as the initial model. Fortunately, the `caret` package (Kuhn, 2023) provides RFE coding with an RF model as the initial model, which is basically an extension of the RT model, as it uses multiple trees instead of just one. As a result, RF was used as the initial model instead of RT.

The first step to create the model in R was to split the training data into a data set that only contained the predictors and a numerical string containing the CLV values, as the rfe() function from the `caret` package (Kuhn, 2023) needs predictors and the dependent variable as separate input data. Before using rfe(), a seed was set for reproduction purposes and rfeControl() was used to set some specifications for RFE. In the rfeControl() function, “functions” was set to “rfFuncs”, “method” was set to “cv” and “number” (number of folds used for cross-validation) was set to “5”. Furthermore, the default values of the parameters were used in the RF model because I could not figure out how to change these parameters using this function. As a result, “ntree” was set to “500” and “mtry” was set to floor(ncol(x)/3). After that, the rfe() function was used to create the RFE model (with RF as the initial model), using the specifications from the rfeControl() function. Moreover, “sizes” was set to “1:31”, so that all available predictors were allowed to be in the optimal subset of features. After performing RFE, the optimal features were obtained by using $optVariables in R. After the optimal subset of features were obtained, the steps explained in section 4.2.4 were repeated. The only difference with those previously explained steps was that only the optimal variables were included in the data, instead of using all predictors.

4.8 Random Forest

The seventh model that was used to predict CLV is an RF model. The first step was to set a seed for reproduction purposes. After that, a preliminary RF model was trained on the training data with the randomForest() function from the `randomForest` package (Breiman et al., 2022). In the randomForest() function, “mtry” was set to the rounded value of the square root of the number of predictors used. Furthermore, “ntree” was set to “700”. The reason for selecting a relatively high number of trees was because the next step was to plot an OOB error plot for the preliminary model. This plot showed the OOB errors for an RF model using ntree values from one to the chosen 700. Using a higher ntree results in a more comprehensive OOB error plot, as more ntree values are included. From the OOB error plot, the ntree values resulting in relatively low OOB errors were selected.

The next step was to tune the hyperparameters used by the RF model. With the selected ntree values and all the mtry values from two to 31, a hyper grid was created with the expand.grid() function in which each combination of a possible mtry value and ntree value was created in a data set. After that, a seed was set for reproduction purposes and the ranger() function from the `ranger` package (Wright et al., 2023) was used to create RF models for all the possible combinations of ntree and mtry values. The combination that resulted in the lowest OOB error was selected for the final RF model. After setting a seed, the final RF model was created with the randomForest() function explained before, but this time using the optimal ntree and mtry value. Also, “importance” was set to “TRUE” so that a permuted variable importance plot could be plotted to improve the interpretability of the model. In short, the permuted variable importance shows the decrease in the model’s accuracy when the values of a certain variable are randomly shuffled. This plot was created with the varImpPlot() where “type” was set to “1” to make sure that the plot showed the permuted variable importance. When “type” is set to “1”, the x-axis shows the mean decrease in accuracy after a single variable is permuted.

When using an RF model, the OOB error can be used to measure the accuracy of the model. However, in this paper, the model accuracy of the (final) RF model needs to be compared with other models that did not use the OOB error as a measurement of accuracy. As a result, to measure the prediction accuracy of the RF model, the predict() function was used to predict the CLV values in the test data. With those values, several accuracy metrics were calculated, which is explained in more detail in section 4.10. The

OOB error was therefore only used to select the optimal number of trees used and the optimal number of random estimators used by the model.

4.9 RF-BSS

The eighth model that was used to predict CLV is an RF-BSS model. Just like the LASSO-BSS model and the RT-BSS model, the BSS step in the RF-BSS model used a linear regression as the initial model, as I could not find a best subset selection algorithm in R that uses a RF as the initial model. As a result, the best subset found in the LASSO-BSS model was used again for the RF-BSS model. With these selected variables instead of using all the predictor variables, the steps explained in section 4.2.7 were repeated. The only slight difference is that this time, the hypergrid was created with mtry values from two to the number of features in the best subset, instead of using mtry values from two to 31.

4.10 RF-RFE

The ninth and final model that was used to predict CLV is an RF-RFE model. The first step was to split the training data into a data set that only contained the predictors and a numerical string containing the CLV values, as RFE needs predictors and the dependent variable as separate input data when using the rfe() function in R. Moreover, rfeControl() was used to set some specifications for RFE. In rfeControl(), "functions" was set to "rfFuncs", "method" was set to "cv", and "number" (number of folds used for cross-validation) was set to "5". Furthermore, rfFuncs used the default values of ntree and mtry (500 and floor(ncol(x)/3) respectively), because I could not figure out how to change these parameters using this function. After that, rfe() was used to create the RFE model with RF as the initial model, using the specifications from the rfeControl() function. Moreover, "sizes" was set to "1:31", so that all available predictors were allowed to be in the optimal subset of features. After performing RFE, the optimal features were obtained by using $optVariables in R. With these selected optimal variables, the steps explained in section 4.2.7 were repeated. The only slight difference is that this time, the hypergrid was created with mtry values ranging from one to the number of optimal features selected by RFE, instead of using mtry values ranging from two to 31.

4.11 Measuring the accuracy of the models

This research used MAE, RMSE and the adjusted R-squared as accuracy metrics to evaluate the predicting performance of the supervised learning models. These metrics were selected because they are easy to interpret and are often used in studies where similar supervised learning models are being used or where similar research questions are being answered (see section 2.9). The reason for using the adjusted R-squared instead of the normal R-squared is because the adjusted R-squared penalizes the inclusion of unnecessary variables in the model, which is favorable because the nine supervised learning models use different numbers of variables. The formulas to calculate MAE, RMSE and the adjusted R-squared are explained below.

MAE can be defined as:

$$MAE = \frac{\sum_{i=1}^{N} |\hat{y}_i - y_i|}{N}$$

where $\hat{y}_i$ represents the predicted value of observation i, y_i represents the observed value of observation i and N represents the sample size. A lower MAE score indicates a higher model accuracy. Moreover, the MAE score can be compared with the range of the dependent variable to see whether the score is sufficient or not.

RMSE can be defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{N}}$$

where $\hat{y}_i$ represents the predicted value of observation i, y_i represents the observed value of observation i, and N represents the sample size. A lower RMSE score indicates a higher model accuracy. Just like the MAE score, the RMSE score can be compared with the range of the dependent variable to see whether the score is sufficient or not.

R-squared can be defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2}$$

where $\hat{y}_i$ represents the predicted value of observation i, y_i represents the observed value of observation i, and $\bar{y}$ represents the mean of the observed values. Additionally, the adjusted R-squared can be defined as:

$$Adjusted\ R^2 = 1 - \frac{1 - (1 - R^2)(N - 1)}{N - p - 1}$$

where N represents the sample size and p represents the number of predictors used. A higher adjusted R-squared indicates a higher proportion of the variance in the dependent variable that can be explained by the predictors.

5 Results

5.1 LASSO, LASSO-BSS and LASSO-RFE

Figure 4 shows how the coefficients of the predictors used in the LASSO regression model change when different values of lambda are used. According to the coefficient paths, *frequency* seems to be the most important variable, followed by *number_months_from_start*, *recency_months*, *registered_via7*, and finally *average_amount_paid*. However, it seems that *frequency*, *number_months_from_start* and *recency_months* are by far the most important predictors. What is also interesting to see is that *registered_via7* seems to be a relatively important variable in the LASSO regression. Unfortunately, it is unknown what type of registration it represents. Moreover, *average_amount_paid* does not seem to be that relevant compared to the other mentioned predictors, which is quite counterintuitive as you would expect it to be the most important predictor together with *frequency*. However, it corresponds with figure 3 in section 3.9, which showed a low correlation between *average_amount_paid* and *CLV*.

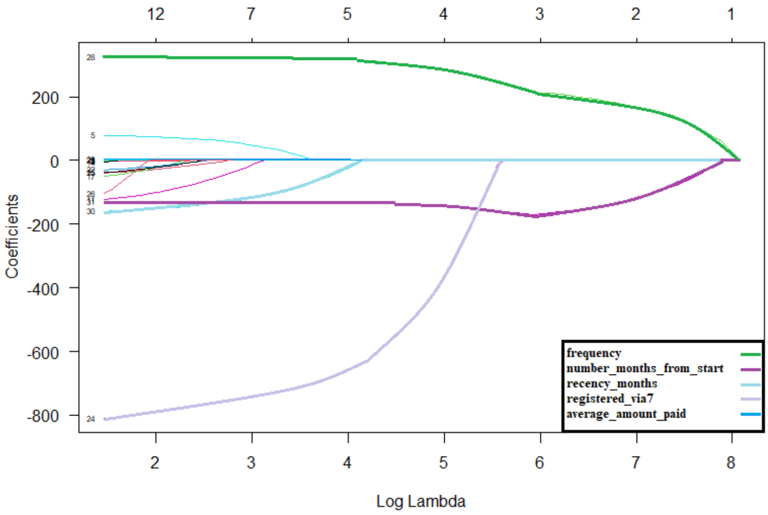

Figure 4 Coefficient paths of the predictors in the LASSO regression model

Figure 5 shows the coefficient paths of the predictors used by the LASSO-BSS model. The best subset of variables used by the LASSO_BSS model included the following 16 variables: *male, bd, city5, city11, city14, city15, city17, city22, registered_via7, registered_via9, registered_via13, is_churn, frequency, average_amount_paid, recency_months* and *number_months_from_start*. What is important to note is that these 16 variables also represent the best subset for the RT-BSS model and the RF-BSS model, because they all use the same BSS method. Figure 4 looks almost identical to figure 3. The only difference between the two figures is that figure 4 contains less coefficient paths because of BSS. However, the coefficient paths of the most important predictors are identical in both figures.

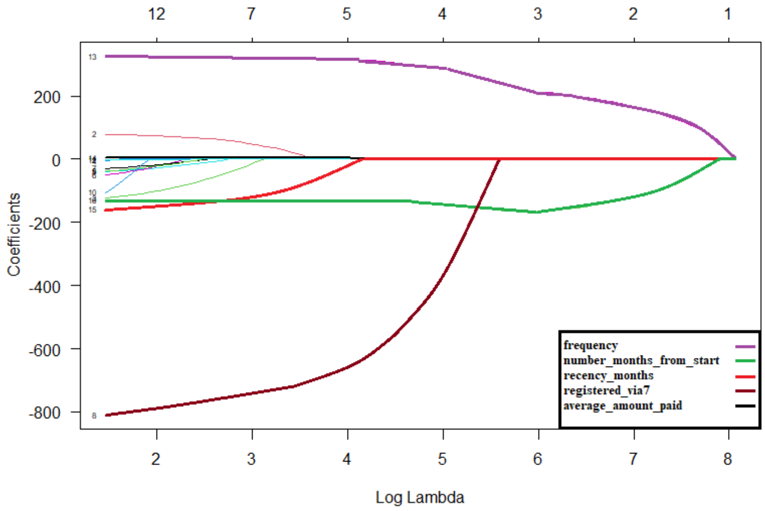

Figure 5 Coefficient paths of the predictors in the LASSO-BSS model

In figure 6, the coefficient paths of the predictors used by the LASSO-RFE model are plotted. In the LASSO-RFE model, all the original 31 predictors were included as optimal features after performing RFE. Figure 6 looks very similar to figure 4 and the coefficient paths of the most important variables are identical except for one variable, which is quite surprising. According to the coefficient paths of figure 6, *frequency* still seems to be the most important variable, but is this time followed by *registered_via9*, *number_months_from_start*, *registered_via7*, and finally *recency_months*.

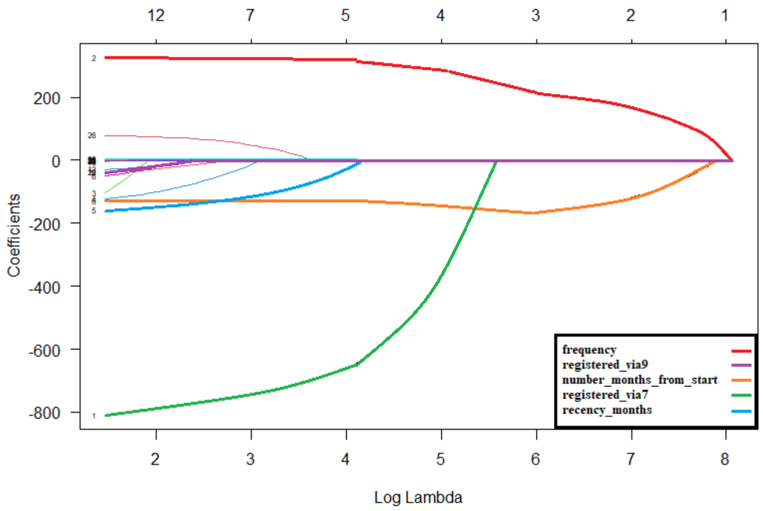

Figure 6 Coefficient paths of the predictors in the LASSO-RFE model

In table 3, the coefficients of the predictors used by the three LASSO model, the LASSO-BSS model and the LASSO-RFE model can be seen. What is interesting to see is that three models are quite similar. They all use an optimal lambda value of 4.32 and use almost the exact same predictors to predict CLV. The regular LASSO regression model uses 18 out of the 31 original predictors to predict CLV, as it shrunk the coefficients of *city6, city7, city8, city9, city10, city12, city13, city16, city18, city19, city20, city21* and *is_churn* to zero. In the LASSO-BSS model, all the previously mentioned city dummy variables were already removed by BSS together *with city3, city4* and *registered_via4*. After that, it only shrunk the coefficient of is_churn to zero. In contrast, the RFE step of the LASSO-RFE model did not remove any variables. From all the 31 remaining features, the LASSO-RFE model shrunk the coefficients of *city3, city6, city7, city8, city9, city10, city12, city16, city18, city19, city20, city21* and *is_churn* to zero. What is interesting to see is that none of the three models either removed *frequency, number_months_from_start, average_amount_paid* or *recency_months* because of the high correlation between *frequency* and *number_months_from_start*, and between *average_amount_paid* and *recency_months*.

Table 3 Coefficients of the predictors used in the final LASSO regression model

Variable	Coefficient LASSO	Coefficient LASSO-BSS	Coefficient LASSO-RFE
intercept	2853.36	2852.16	2854.07
male	-2.30	-2.30	-1.07
bd	-1.01	-1.01	-2.31
city3	-0.83	x	0.00
city4	0.70	x	1.14
city5	79.15	78.75	79.37
city6	0.00	x	0.00
city7	0.00	x	0.00
city8	0.00	x	0.00
city9	0.00	x	0.00
city10	0.00	x	0.00
city11	-123.16	-123.58	-122.12
city12	0.00	x	0.00
city13	0.00	x	0.43
city14	-3.81	-4.06	-2.93
city15	-38.52	-38.74	-37.40
city16	0.00	x	0.00
city17	-49.74	-49.97	-48.62
city18	0.00	x	0.00
city19	0.00	x	0.00
city20	0.00	x	0.00
city21	0.00	x	0.00
city22	-28.96	-29.15	-27.84
registered_via4	0.70	x	1.02
registered_via7	-811.10	-811.18	-810.44
registered_via9	-38.46	-38.54	-38.58
registered_via13	-105.38	-105.58	-105.51
is_churn	0.00	0.00	0.00
frequency	326.54	326.54	326.44
average_amount_paid	4.55	4.55	4.55
recency_months	-159.52	-159.49	-160.19
number_months_from_start	-130.04	-130.03	-130.14

Note: “x” means that the variable is removed by BSS or RFE.

5.2 RT, RT-BSS and RT-RFE

In figure 7, a visualization of the RT model can be seen. What is important to note is that this figure only contains the first few splits of the three model instead of the entire tree. This is because the full pruned tree, which uses a cp of “2.634944e-08”, is too large to be visualized together with information about which tests are used at the splits of the tree, even after pruning. However, in figure A3 and A4 in the appendix, the structure (without additional information about the tests used) of the full tree before and after CCP can be seen for a better understanding of the structure of the model.

When looking at the first few splits of the RT model in figure 7, it becomes clear that the most important three variables used by the RT model to predict CLV are *number_months_from_start* followed by *frequency* and *average_amount_paid* respectively. This is quite an interesting result, as it would be more intuitive if *frequency* was more important than *number_months_from_start*, based on the formula how CLV was calculated. However, this unintuitive finding is likely caused by multicollinearity. Additionally, what is interesting to see is that both variables are included by the model which shows that the RT model is not good at handling multicollinearity.

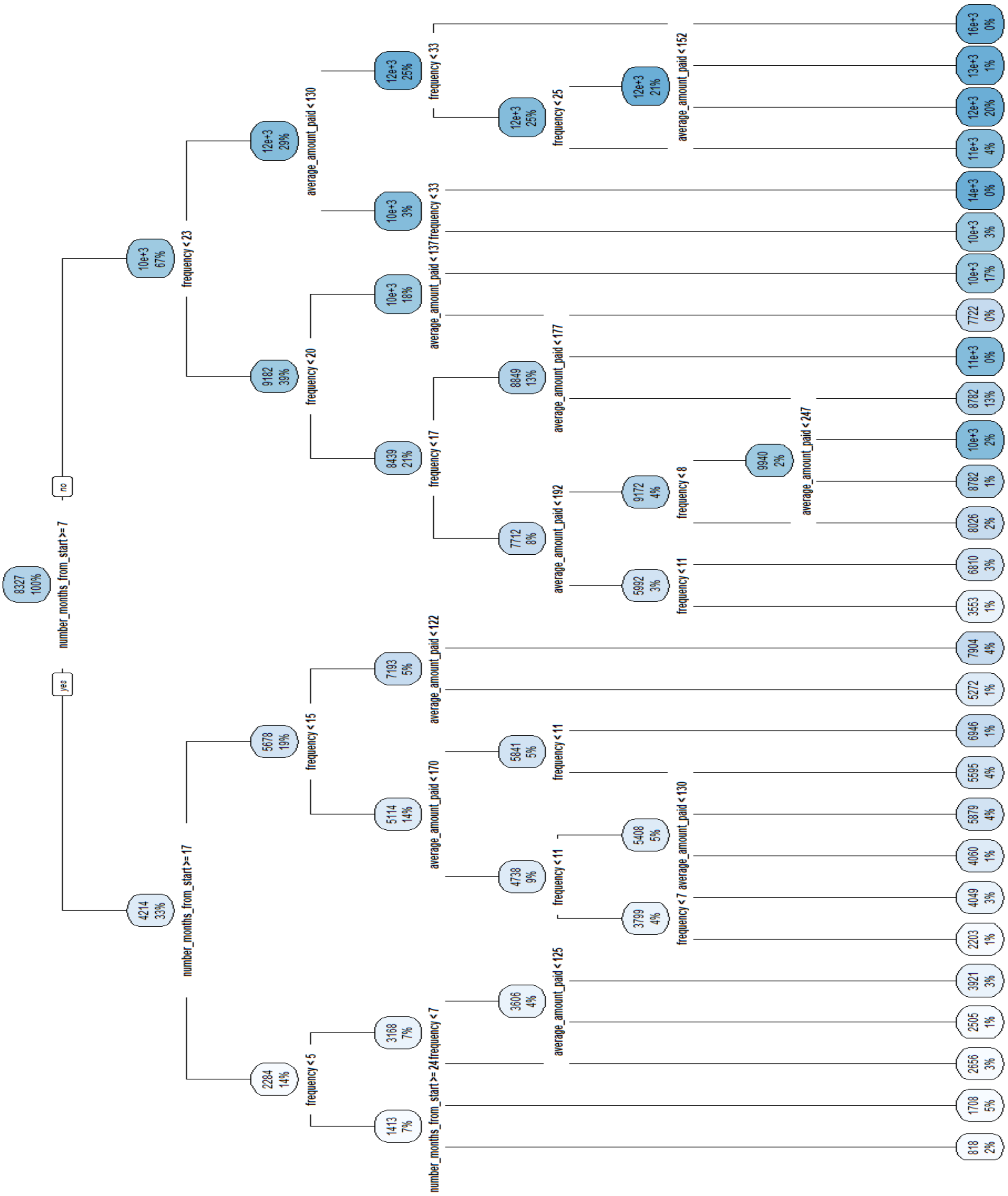

Figure 7 Visualization of the first few splits of the RT model

In figure 8, a visualization of the RT-BSS model can be seen. Just like figure 7, it only contains the first few splits of the tree because of the large size of the full tree used in the analysis. However, in figure A5 and A6 in the appendix, the structure of the full tree before and after CCP, which used a cp value equal to "8.992613e-08", can be seen for a better understanding of the structure of the model. Figure 8 is identical to figure 7. Just like in the RT model, the most important three variables used by the RT-BSS model to predict CLV are *number_months_from_start* followed by *frequency* and *average_amount_paid* respectively.

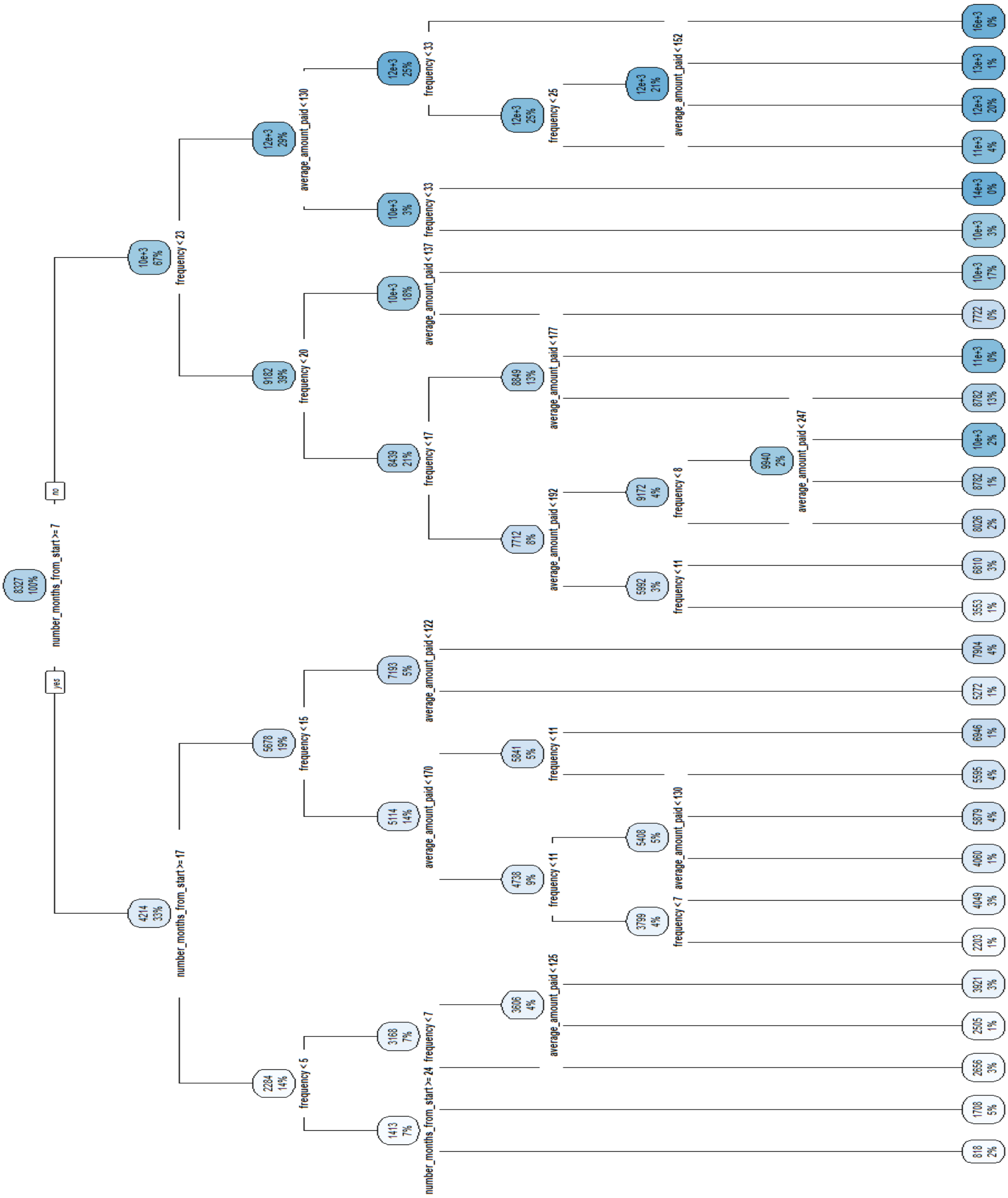

Figure 8 Visualization of the first few splits of the RT-BSS model

In figure 9, a visualization of the RT-RFE model can be seen. Just like figure 7 and figure 8, it only contains the first few splits of the tree because of the large size of the full tree used in the analysis. However, the structure of the tree before and after CCP, which used a cp value equal to “2.340427e-10”, can be seen in figure A7 and A8 in the appendix for a better understanding of the structure of the model. In the RT-RFE model, only *frequency* and *average_amount_paid* were selected as the optimal features. As a result, the tree only uses these two variables to predict CLV. However, figure 9 shows that *frequency* is the most important out of the two. These findings are also quite intuitive.

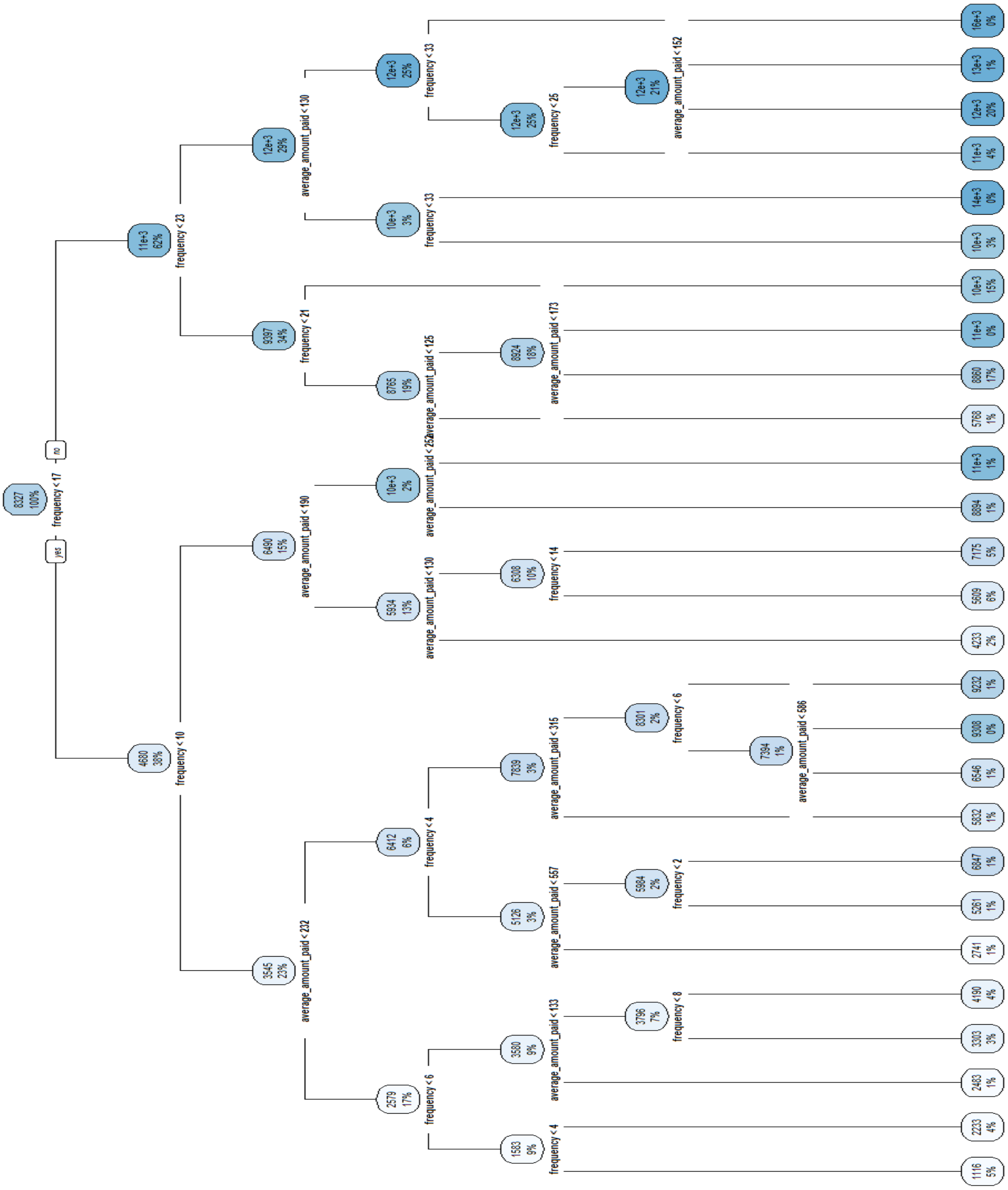

Figure 9 Visualization of the first few splits of the RT-RFE model

5.3 RF, RF-BSS and RF-RFE

Figure 10 shows the OOB error plot for the preliminary RF model, which used six random predictors to build each tree. For this model, 280, 240, 350, 400, 475, 550 and 650 seemed to be the most interesting values for the number of trees used by the model, as these values resulted in the lowest OOB errors for the RF model. a result, these ntree values together with the mtry values ranging from two to 31 were used in the grid search to find the optimal hyperparameter values for the final RF model.

What is important to note is that the plot was initially used to identify the best ntree values on a different scale. However, after determining the best optimal ntree values, the Y-axis scale was adjusted for the plot so that it would have the same Y-axis dimensions as the OOB error plot of the preliminary RF-BSS model and the preliminary RF-RFE model shown further below. As a result, it may seem as though the selected ntree values do not result in the lowest OOB errors within the newly defined Y-axis range. This is also the case for figure 11 and 12, which are explained further below.

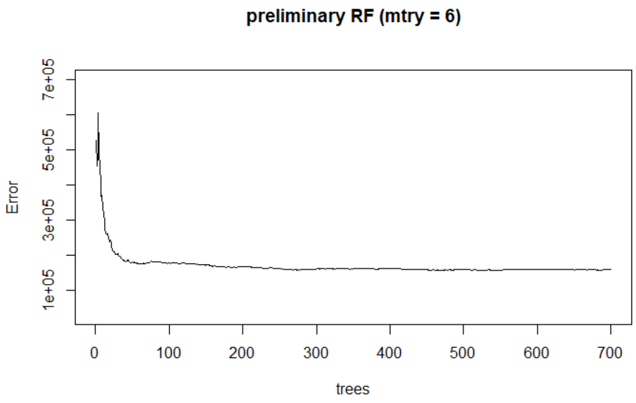

Figure 10 OOB error plot of the preliminary RF model

Figure 11 shows the OOB error plot for the preliminary RF-BSS model, which used four random predictors to build each tree, because the best subset used by the model included 16 variables. For this model, 70, 200, 320, 390, 515 and 700 were selected as the most interesting ntree values and were used to create the grid search together with mtry values ranging from two to 31.

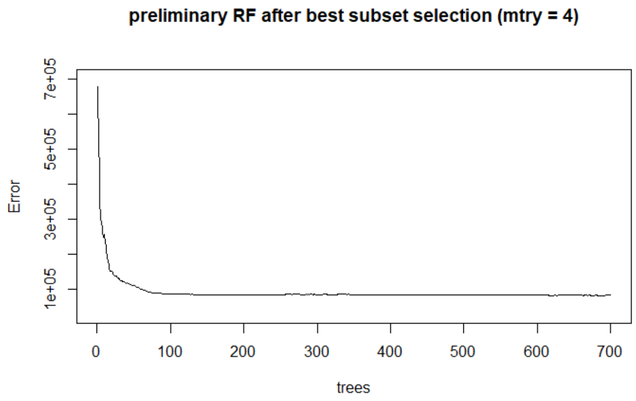

Figure 11 OOB error plot of the preliminary RF-BSS model

Figure 12 shows the OOB error plot for the preliminary RF-RFE model. The model only used one random predictor to build each tree, because RFE only selected *frequency* and *average_amount_paid* as the optimal values to be included in the model. For the RF-RFE model, 200, 300, 370, 500, 600 and 700 were selected as the most interesting ntree values and were used to create the grid search together with mtry values ranging from one to two.

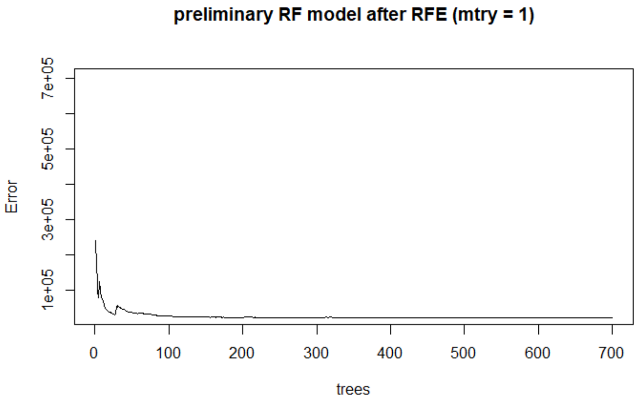

Figure 12 OOB error plot of the preliminary RF-BSS model

The grid search conducted for the RF model showed that the optimal number of random variables used in each tree was 31 and the optimal number of trees used by the model was 550. Moreover, the grid search conducted for the RF-BSS model showed that the optimal number of random variables used in each tree was 16 and the optimal number of trees used by the model was 390. Finally, the grid search conducted for the RF-BSE model showed that the optimal number of random variables used in each tree was 2 and the optimal number of trees used by the model was 500. These results are quite interesting, as it suggests that using all available variables instead of using a random subset for each tree results in a higher prediction accuracy. In other words, the results seem to suggest that a bagging model would predict CLV more accurately compared to a RF model. This is unexpected, because according to Breiman (2001), the variable randomness should minimize correlation while still maintaining strength.

In figure 13, the permuted variable importance of the RF model can be seen. According to Breiman et al. (2022), the X-axis shows the mean decrease in accuracy. However, this does not seem to be in line with the actual X-axis in figure 13 (or figure 14 and 15). As a result, the figure should be interpreted carefully. However, what is clear from it is that the most important variable to predict CLV for the RF model is *frequency*, followed by *average_amount_paid* and *number_months_from_start*. The remaining predictors are not relevant when predicting CLV. The variable importance order is also very intuitive and looks similar to the importance order of the LASSO and the LASSO-BSS model. However, it is interesting to see that all of the highly correlated variables mentioned in section 3.9 are seen as the most important variables, which seems to suggest that the basic RF model does not remove correlated variables to prevent multicollinearity. Moreover, it is interesting that *average_amount_paid* is a very important predictor for the RF model when predicting CLV, while there is a very weak correlation between *average_amount_paid* and *CLV*.

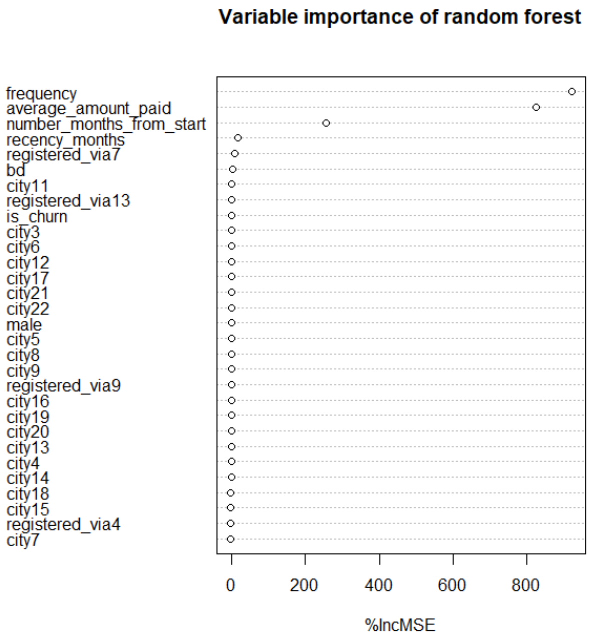

Figure 13 Permuted variable importance of the RF model

In figure 14, the permuted variable importance of the RF-BSS model can be seen. The variable importance order of the RF-BSS model is very similar to the variable importance order of the RF model for the first few variables. Again, *frequency*, *average_amount_paid*, *number_months_from_start*, are the most important variables to predict CLV. The remaining predictors are not relevant when predicting CLV. What is interesting to see is that *average_amount_paid* is a very important predictor for the RF-BSS model when predicting CLV, while there is a very weak correlation between *average_amount_paid* and *CLV.*

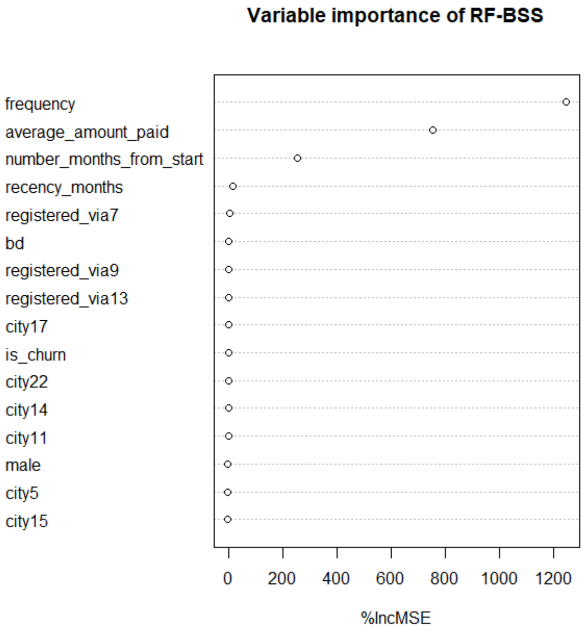

Figure 14 Permuted variable importance of the RF-BSS model

In figure 15, the permuted variable importance of the RF-RFE model can be seen. The figure shows that that *frequency* is the most important variable, but *average_amount_paid* is still quite relevant. Just like the RF model and the RF-BSS model, it needs both variables for a good prediction. This is also very intuitive. However, what is interesting to see is that *average_amount_paid* is a very important predictor for the RF-RFE model when predicting CLV, while there is a very weak correlation between *average_amount_paid* and *CLV.*

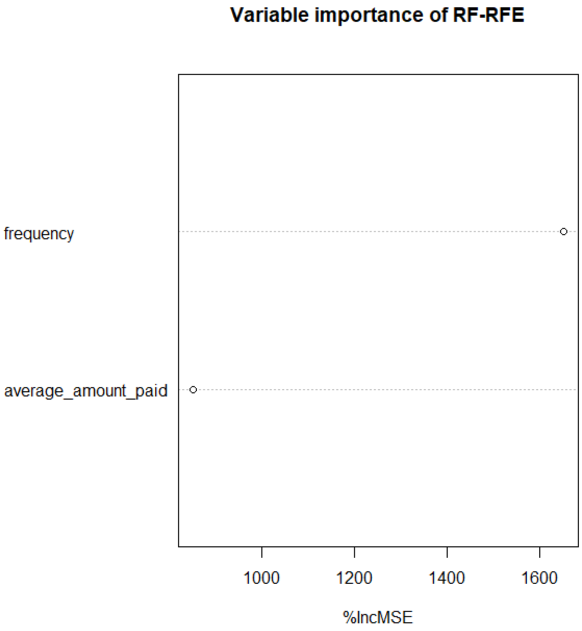

Figure 15 Permuted variable importance of the RF-RFE model

5.4 Prediction accuracies

In table 4, the accuracy metrics of the nine supervised learning models can be seen. In general, the table shows that based on the MAE, RMSE and adjusted R-squared, the RF type models have the highest prediction accuracy, followed by the RT type models and finally the LASSO regression type models. What is interesting to see is that the RT type models have a relatively high prediction accuracy, which seems to contradict the literature as it suggested that RT models often have poor prediction accuracy. However, a reason why the models perform so well, might be because the relationship between the most important predictors and *CLV* is quite simple.

When looking at the prediction accuracies in a bit more detail, it becomes clear that based on the three accuracy measures, the best performing supervised learning model is the RF-RFE model, followed by both the RF-BSS model and the RF model, followed by the RT-RFE model, the RT-BSS model, the RT model, the LASSO-RFE model, the LASSO-BSS model, and finally the LASSO regression model. What is interesting to see is that BSS improves the prediction accuracy for the RT model, whereas its impact on the LASSO regression model and the RF model appears to be limited. For the LASSO regression model, it slightly increases MAE and RMSE, but also increases the adjusted R-squared, while it slightly increases the adjusted R-squared and does not seem to influence MAE or RMSE. On the other hand, RFE seems to improve the prediction accuracy of all three types of supervised learning models. However, for the LASSO regression model, the increase in accuracy is very small.

The nine different supervised learning models all seem to accurately predict CLV. They all have an adjusted R-squared that is higher than 0.93, which is quite high. The MAE and RMSE values also seem to be quite low; especially given the fact that CLV ranges from 156.58 to 18,376.77.

Table 4 Prediction accuracies of the supervised learning models

model	MAE	RMSE	Adjusted R^2
LASSO	570.59	855.60	0.939612
LASSO-BSS	570.60	855.61	0.939691
LASSO-RFE	570.58	855.55	0.939620
RT	41.312	149.93	0.998146
RT-BSS	41.22	149.77	0.998152
RT-RFE	22.85	91.75	0.999308
RF	16.72	88.20	0.999358
RF-BSS	16.72	88.20	0.999359
RF-RFE	4.93	32.61	0.999913

Note: The results in the third column of this table have more digits compared the first two columns to properly show the slight differences between the adjusted R-squared values of the supervised learning models.

6 Conclusion

The central research question that was formulated during this research was: *"What type of supervised learning model is most suitable for predicting CLV when using music streaming subscription data?"* However, as stated before in the introduction, the three sub research questions need to be answered first to answer the central research question. The sub questions of this research are:

"How does the model interpretability vary for a LASSO regression, a regression tree and a random forest when predicting CLV?"

"How does the prediction accuracy vary for a LASSO regression, a regression tree and a random forest, when predicting CLV?"

"How does the prediction accuracy of each model change when recursive feature elimination or best subset selection is used in addition?"

6.1 Model interpretability

When assessing the interpretability of LASSO regression, an RT, and an RF for CLV prediction, distinctive characteristics were found for each of these types of models. The LASSO type models that were used in this research offered clear insights into which variables were used and how they affected CLV through their respective coefficients, aided by the coefficient paths that illustrated how the coefficients of the predictor variables responded to various lambda values, which gave an indication about the variable importance order.

RTs are typically very easy to interpret by looking at the visualized plot of the tree model. Unfortunately, the RT models that were used in this research used large trees that did not fit in one plot when including information about the test used at each split. Consequently, only the first few splits of the tree could be plotted with this additional information. As a result, it was only evident which variables held the highest level of importance. Consequently, due to this additional information, only the initial tree splits could be graphed, leading to ambiguity regarding the predictors utilized in subsequent splits. As a result, the relevance of variables on which tests were founded was only evident in the initial splits.

Finally, RF models are generally very hard to interpret, because they are black-box models that do not explain their decision-making process. For example, it was not possible to see the multiple individual trees used by the RF models in this research. As a result, it was unknown which variables were used for prediction in each tree, what tests were used and in what order these tests were used. However, the RF-RFE model was an exception to this rule as it only used two relevant predictors. Consequently, it was obvious that the RF-RFE model used these two variables, but it was still not clear which tests were used and in what order they were used. Fortunately, the variable importance plot improved the interpretability of the RF models by showing how important each variable was relative to each other to predict CLV.

In conclusion, LASSO regression is the most interpretable type of model compared to an RT and an RF when predicting CLV. An RT (with complex trees) comes next in interpretability, followed by an RF, which proved to be the least interpretable in this research. However, because the RT models used complex trees, the interpretability of an RF was only slightly worse in this research.

6.2 Model accuracy

When assessing how the prediction accuracy varied for a LASSO regression, an RT, and an RF, when predicting CLV, it became clear that the RF had the highest prediction accuracy, followed by the RT and finally the LASSO regression. What is interesting to see is that the RT and the RF performed much better compared to the LASSO regression. This might suggest that there is a non-linear relationship between the predictors and *CLV*, as an RT and an RF can both capture non-linear relations, while the LASSO regression is not designed for this. However, the LASSO regression still had a good prediction accuracy, which seems to suggest that fitting a straight line also works relatively well.

6.3 The effect of additional feature selection methods on accuracy

When assessing how the accuracy of each model changed when RFE or BSS was used as an additional feature selection method, it became clear that RFE increased the accuracy of the three types of supervised learning models across the various accuracy metrics used in this research. However, for the LASSO regression model, the increase in accuracy was very small. Also, RFE did not filter out any variables. As a result, the LASSO-RFE model used the exact same variables in the regression as the regular LASSO model when using the same seed. The only difference between the models was that the variables in the data set used by the LASSO-RFE model were ordered differently. Somehow, this resulted in a slight improvement in terms of prediction accuracy. On the other hand, the increase in prediction accuracy was very large for the RT and the RF when RFE was used as an additional feature selection method. One possible explanation for this phenomenon could be that the LASSO regression model is a proper feature selection method by itself. Consequently, combining RFE with a supervised learning model that already possesses a built-in feature selection capability may not increase prediction accuracy as much as when RFE is combined with a supervised learning model that lacks such capabilities.

Besides RFE, adding BSS only seemed to slightly improve the prediction accuracy for the RT model based on the three accuracy metrics. Moreover, Adding BSS did not seem to increase MAE or RMSE for the RF model, but it slightly increased the adjusted R-squared because it uses 16 predictors instead of 31. Finally, Adding BSS to the LASSO regression increased MAE and RMSE, but it increased the adjusted R-squared, because LASSO-BSS uses 16 variables instead of 31. Hence, it appears that BSS has a minimal impact on the prediction accuracy of the LASSO regression, RT, and RF models.

6.4 The most suitable supervised learning model to predict CLV

Based on the answers to the sub questions, the most suitable supervised learning model to predict CLV seems to be the RF-RFE model. In this research, it clearly had the best prediction accuracy based on MAE, RMSE and adjusted R-squared, and it only used the two most intuitive predictors to predict CLV, namely *frequency* and *average_amount_paid*. As a result, the model can be interpreted relatively well for an RF model. Therefore, in my opinion, using an RF-RFE model would be the most suitable model for companies to predict CLV when it is initially calculated with the basic CLV calculation method. However, if a company prioritizes interpretability and they would reject the RF-RFE model, I suggest to opt for the RT-RFE model instead. It utilizes the same two intuitive predictors, offers improved interpretability, and is still very accurate.

7 Discussion

7.1 Research limitations

This research also has a few limitations. First, a few highly correlated variables were used as predictors, which seemed to cause multicollinearity in some of the supervised learning models. As a result, the true effects of some of the variables might be different compared to the observed effects. Moreover, there is a very small negative correlation between *average_amount_paid* and *CLV*, while a stronger positive correlation would be more intuitive. This might have been caused by a few users in the data that had a subscription with an annual contract. These users were often subscribed to the service for only one year, and paid very high fees, as they had to pay for the entire year at once. This would also clarify why *average_amount_paid* is less relevant when predicting CLV than what is expected.

Additionally, in this research, CLV was calculated with the basic CLV calculation method. This method seems to be suitable for CLV calculation in contractual settings, as it tries to predict future revenues based on historical purchases and multiplies the revenues with the average retention rate to account for churn. The only thing that the method does not account for in a contractual setting are changes in fees, such as user upgrades or downgrades in subscriptions, which may affect CLV. However, a lot of the users did not behave like you would expect them to behave in a contractual setting as they often unsubscribed and resubscribed to the service. As a result, it might have been better to estimate CLV with the Pareto/NBD model or NG/NBD model, because these models are able to predict irregular individual customer purchase behavior. Moreover, because of this strange behavior, it was hard to determine when a user had churned or not, which might have resulted in a biased retention rate.

7.2 Recommendations for future research

For future research, I have several recommendations. First, I would recommend conducting this research again and exclude variables that might cause multicollinearity. Moreover, I would advise to include additional predictor variables into the analysis to explore whether there are other predictors capable of predicting CLV. This research has primarily showed the relevance of *frequency*, *average_amount_paid*, and variables closely correlated with these factors when predicting CLV. However, there could be other relevant features, such as customer satisfaction, that are relevant when predicting CLV. Furthermore, I would recommend creating a second dependent CLV variable that is estimated with the Pareto/NBD model or NG/NBD model. By adopting this approach, it can be examined how dissimilar both differently calculated CLV variables are across both lower and higher values. Besides that, it can be tested whether one of the two CLV variables is more predictable than the other.

Additionally, I would advise to use a different statistic for BSS that results in a smaller subset, as using the adjusted R-squared resulted in BSS having minimal impact on the prediction accuracies of the LASSO regression, the RT, and the RF. Moreover, I would advise to add a more comprehensive feature selection method alongside BSS and RFE that generally removes more features than RFE and BSS. Marketeers often include all available features and interactions between features in supervised learning to capture all relationships between the predictors and the dependent variable. However, simpler models often yield greater interpretability, reduced computation time, and, in some instances, improved prediction accuracy. Therefore, using three different feature selection methods, each resulting in varying subset sizes, effectively demonstrates the impact of model simplicity on interpretability and prediction accuracy for different supervised learning models. Besides that, it would be interesting to see whether a more comprehensive feature selection would further improve the prediction accuracy more effectively than RFE.

Finally, I would advise to create a new RFE package that is capable of using a LASSO regression, an RT, other widely-adopted supervised learning models as the primary models for the RFE process, as the currently used caret package in R lacks this functionality.

When the previous recommendations would all be incorporated in future research, my final recommendation would be to conduct the research on multiple customer/user data sets from different industries, to see whether results are robust.

8 References

Ai, C. (2022). A method for cancer genomics feature selection based on LASSO-RFE. Iranian Journal of Science and Technology, Transactions A: Science, 46(3), 731-738.

Akins, R. B., Tolson, H., & Cole, B. R. (2005). Stability of response characteristics of a Delphi panel: application of bootstrap data expansion. *BMC medical research methodology, 5*(1), 1-12.

Algamal, Z. Y., & Lee, M. H. (2015). Adjusted adaptive lasso in high-dimensional poisson regression model. *Modern Applied Science, 9*(4), 170.

Anguita, D., Ghelardoni, L., Ghio, A., Oneto, L., & Ridella, S. (2012). The'K'in K-fold Cross Validation. In ESANN (pp. 441-446).

Antipov, E. A., & Pokryshevskaya, E. B. (2012). Mass appraisal of residential apartments: An application of Random forest for valuation and a CART-based approach for model diagnostics. *Expert systems with applications, 39*(2), 1772-1778.

Asadi, N., & Kazerooni, M. (2023). A stacked ensemble learning method for customer lifetime value prediction. *Kybernetes*.

Barrowman, M., PhD. (2020). Convert all Character variables to Factors. *MyKo101*. https://michaelbarrowman.co.uk/post/convert-all-character-variables-to-factors/#:~:text=Why%20factors%3F,we're%20going%20to%20get.

Bayam, E., Liebowitz, J., & Agresti, W. (2005). Older drivers and accidents: A meta analysis and data mining application on traffic accident data. *Expert Systems with Applications, 29*(3), 598-629.

Bel, L., Allard, D., Laurent, J. M., Cheddadi, R., & Bar-Hen, A. (2009). CART algorithm for spatial data: Application to environmental and ecological data. *Computational Statistics & Data Analysis, 53*(8), 3082-3093.

Berger, P. D., & Nasr, N. I. (1998). Customer lifetime value: Marketing models and applications. *Journal of interactive marketing, 12*(1), 17-30.

Bertsimas, D., King, A., & Mazumder, R. (2016). Best subset selection via a modern optimization lens.

Bhalla, D. (2016). *R : Converting Multiple Numeric Variables to Factor*. ListenData. https://www.listendata.com/2015/05/converting-multiple-numeric-variables.html#:~:text=In%20R%2C%20categorical%20variables%20need,them%20as%20a %20grouping%20variable

Blattberg, R. C., Kim, B., & Neslin, S. A. (2009). *Database Marketing: Analyzing and Managing Customers*. Springer.

Bonacchi, M., & Perego, P. (2012). Measuring and managing customer lifetime value: A CLV scorecard and cohort analysis in a subscription-based enterprise. *Management Accounting Quarterly, 14*(1), 27.

Botchkarev, A. (2018). Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. *arXiv preprint arXiv:1809.03006*.

Bradford, J. P., Kunz, C., Kohavi, R., Brunk, C., & Brodley, C. E. (1998). Pruning decision trees with misclassification costs. In *Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21–23, 1998 Proceedings 10* (pp. 131-136). Springer Berlin Heidelberg

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5-32.

Breiman, L., Cutler, A., Liaw, A., & Wiener, M. (2022). *CRAN - Package RandomForest*. CRAN. https://cran.r-project.org/web/packages/randomForest/index.html

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). Cart. *Classification and regression trees*.

Brownlee, J. (2019). How to choose a feature selection method for machine learning. Machine Learning Mastery, 10.

Burelli, P. (2019). Predicting customer lifetime value in free-to-play games. In *Data analytics applications in gaming and entertainment* (pp. 79-107). Auerbach Publications.

Burkart, N., & Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, *70*, 245-317.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?– Arguments against avoiding RMSE in the literature. *Geoscientific model development, 7*(3), 1247-1250.

Chang, W., Chang, C., & Li, Q. (2012). Customer lifetime value: A review. Social Behavior and Personality: an international journal, 40(7), 1057. Chen, X. W., & Jeong, J. C. (2007, December). Enhanced recursive feature elimination. In *Sixth international conference on machine learning and applications (ICMLA 2007)* (pp. 429-435). IEEE.

Chen, X. W., & Jeong, J. C. (2007, December). Enhanced recursive feature elimination. In *Sixth international conference on machine learning and applications (ICMLA 2007)* (pp. 429-435). IEEE.

Cheng, C. H., & Chen, Y. S. (2009). Classifying the segmentation of customer value via RFM model and RS theory. Expert systems with applications, 36(3), 4176-4184.

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, *7*, e623.

Ching, W. K., Ng, M. K., Wong, K. K., & Altman, E. (2004). Customer lifetime value: stochastic optimization approach. *Journal of the Operational Research Society*, *55*(8), 860-868.

Dahana, W. D., Miwa, Y., & Morisada, M. (2019). Linking lifestyle to customer lifetime value: An exploratory study in an online fashion retail market. Journal of Business Research, 99, 319-331.

Darst, B. F., Malecki, K. C., & Engelman, C. D. (2018). Using recursive feature elimination in random forest to account for correlated variables in high dimensional data. BMC genetics, 19(1), 1-6.

De Myttenaere, A., Golden, B., Le Grand, B., & Rossi, F. (2015). Using the Mean Absolute Percentage Error for Regression Models. In *ESANN*.

De Myttenaere, A., Golden, B., Le Grand, B., & Rossi, F. (2016). Mean absolute percentage error for regression models. *Neurocomputing*, *192*, 38-48.

Desai, S., & Ouarda, T. B. (2021). Regional hydrological frequency analysis at ungauged sites with random forest regression. *Journal of Hydrology*, 594, 125861.

Efron, B. (1979). Computers and the theory of statistics: thinking the unthinkable. *SIAM review*, *21*(4), 460-480.

Efron, B., & Tibshirani, R. (1991). Statistical data analysis in the computer age. *Science, 253*(5018), 390-395.

Fader, P. S., & Hardie, B. G. (2007). How to project customer retention. Journal of Interactive Marketing, 21(1), 76-90.

Fader, P. S., Hardie, B. G., & Lee, K. L. (2005A). RFM and CLV: Using iso-value curves for customer base analysis. *Journal of marketing research*, *42*(4), 415-430.

Fader, P. S., Hardie, B. G., & Lee, K. L. (2005B). “Counting your customers” the easy way: An alternative to the Pareto/NBD model. *Marketing science*, *24*(2), 275-284.

Foss, T., Stensrud, E., Kitchenham, B., & Myrtveit, I. (2003). A simulation study of the model evaluation criterion MMRE. *IEEE Transactions on software engineering*, *29*(11), 985-995.

Friedman, J. H., & Meulman, J. J. (2003). Multiple additive regression trees with application in epidemiology. *Statistics in medicine*, *22*(9), 1365-1381.

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation.

Glady, N., Baesens, B., & Croux, C. (2009A). Modeling churn using customer lifetime value. *European Journal of Operational Research*, *197*(1), 402-411.

Glady, N., Baesens, B., & Croux, C. (2009B). A modified Pareto/NBD approach for predicting customer lifetime value. *Expert Systems with Applications*, *36*(2), 2062-2071.

Gomes, C. M. A., Amantes, A., & Jelihovschi, E. G. (2020). Applying the regression tree method to predict students’ science achievement. *Trends in Psychology*, *28*(1), 99-117.

Gomes, C. M. A., & Jelihovschi, E. (2020). Presenting the regression tree method and its application in a large-scale educational dataset. *International Journal of Research & Method in Education*, *43*(2), 201-221.

Gupta, S., Hanssens, D., Hardie, B., Kahn, W., Kumar, V., Lin, N., ... & Sriram, S. (2006). Modeling customer lifetime value. Journal of service research, 9(2), 139-155.

Gurău, C., & Ranchhod, A. (2002). Measuring customer satisfaction: a platform for calculating, predicting and increasing customer profitability. *Journal of Targeting, Measurement and Analysis for Marketing*, *10*, 203-219.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. Machine learning, 46, 389-422.

Haenlein, M., Kaplan, A. M., & Beeser, A. J. (2007). A model to determine customer lifetime value in a retail banking context. *European Management Journal*, *25*(3), 221-234.

Hansen, L. P., & Jagannathan, R. (1997). Assessing specification errors in stochastic discount factor models. The Journal of Finance, 52(2), 557-590.

Hastie, T., Tibshirani, R., & Tibshirani, R. (2020). Best subset, forward stepwise or lasso? Analysis and recommendations based on extensive comparisons.

Henrard, S., Speybroeck, N., & Hermans, C. (2015). Classification and regression tree analysis vs. multivariable linear and logistic regression methods as statistical tools for studying haemophilia. *Haemophilia*, *21*(6), 715-722.

Hocking, R. R., & Leslie, R. N. (1967). Selection of the best subset in regression analysis. *Technometrics*, *9*(4), 531-540.

Hu, Y. H., Huang, T. C. K., & Kao, Y. H. (2013). Knowledge discovery of weighted RFM sequential patterns from customer sequence databases. Journal of systems and software, 86(3), 779-788.

Hughes, A. M. (2005). *Strategic database marketing*. McGraw-Hill Pub. Co..

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008). Random survival forests.

Jawale, K. (2022). Why Your Subscription Management Solution Must Be Built on Your CRM. *Work 365 Apps*. https://www.work365apps.com/why-your-subscription-management-solution-must-be-built-on-your-crm-and-not-the-accounting-system/

Jeon, H., & Oh, S. (2020). Hybrid-recursive feature elimination for efficient feature selection. Applied Sciences, 10(9), 3211.

Jović, A., Brkić, K., & Bogunović, N. (2015, May). A review of feature selection methods with applications. In 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO) (pp. 1200-1205). Ieee.

Kahreh, M. S., Tive, M., Babania, A., & Hesan, M. (2014). Analyzing the applications of customer lifetime value (CLV) based on benefit segmentation for the banking sector. *Procedia-Social and Behavioral Sciences*, *109*, 590-594.

Kim, J., Nam, C., & Ryu, M. H. (2017). What do consumers prefer for music streaming services?: A comparative study between Korea and US. Telecommunications Policy, 41(4), 263-272.

KKBOX. (2023). About KKBOX - KKBOX. https://www.kkbox.com/about/tw/en

Kotler, P. (1974). Marketing during periods of shortage. *Journal of marketing*, *38*(3), 20-29.

Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, 39, 261-283.

Kuhn, M. (2023). *Classification and Regression Training [R package caret version 6.0-94]*. CRAN. https://cran.r-project.org/web/packages/caret/index.html

Kumar, V., Venkatesan, R., Bohling, T., & Beckmann, D. (2008). Practice Prize Report—The power of CLV: Managing customer lifetime value at IBM. *Marketing science*, *27*(4), 585-599.

Larivière, B., & Van den Poel, D. (2005). Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert systems with applications, 29*(2), 472-484.

Lawrence, R. L., & Wright, A. (2001). Rule-based classification systems using classification and regression tree (CART) analysis. *Photogrammetric engineering and remote sensing*, *67*(10), 1137-1142.

Li, J. (2017). Assessing the accuracy of predictive models for numerical data: Not r nor r2, why not? Then what?. *PloS one*, *12*(8), e0183250.

Loh, W. Y. (2011). Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *1*(1), 14-23.

Lumley, T. (2020). *CRAN - Package leaps*. CRAN. https://cran.r-project.org/web/packages/leaps/index.html

Malthouse, E. C., & Blattberg, R. C. (2005). Can we predict customer lifetime value?. *Journal of interactive marketing*, *19*(1), 2-16.

Milborrow, S. (2022). *CRAN - Package RPart.plot*. CRAN. https://cran.r-project.org/web/packages/rpart.plot/index.html

Ouedraogo, I., Defourny, P., & Vanclooster, M. (2018). Application of random forest regression and comparison of its performance to multiple linear regression in modeling groundwater nitrate concentration at the African continent scale. *Hydrogeology Journal*.

PCMag. (2023). Definition of music streaming service. PCMAG. https://www.pcmag.com/encyclopedia/term/music-streaming-service

Petrison, L. A., Blattberg, R. C., & Wang, P. (1997). Database marketing: Past, present, and future. *Journal of Direct Marketing*, *11*(4), 109-125.

Pfeifer, P. E., & Carraway, R. L. (2000). Modeling customer relationships as Markov chains. *Journal of interactive marketing, 14*(2), 43-55.

Prodromidis, A. L., & Stolfo, S. (1998). Pruning classifiers in a distributed meta-learning system.

Prodromidis, A. L., & Stolfo, S. J. (2001). Cost complexity-based pruning of ensemble classifiers. *Knowledge and Information Systems*, 3, 449-469.

Qiu, X., Fu, D., Fu, Z., Riha, K., & Burget, R. (2011). The method for material corrosion modelling and feature selection with SVM-RFE. In *2011 34th International Conference on Telecommunications and Signal Processing (TSP)* (pp. 443-447). IEEE.

Radečić, D. (2022). Time Series from scratch — Train/Test splits and evaluation metrics. *Medium*. https://towardsdatascience.com/time-series-from-scratch-train-test-splits-and-evaluation-metrics-4fd654de1b37#:~:text=Train%2Ftest%20splits%20in%20time%20series,-In%20machine%20learning&text=For%20example%2C%20if%20you%20had,(2%20years)%20for%20testing.&text=And%20that's%20all%20there%20is%20to%20train%2Ftest%20splits.

Ranstam, J., & Cook, J. A. (2018). LASSO regression. Journal of British Surgery, 105(10), 1348-1348.

Reinartz, W. J., & Kumar, V. (2003). The impact of customer relationship characteristics on profitable lifetime duration. *Journal of marketing, 67*(1), 77-99.

Roy, S. S., Mittal, D., Basu, A., & Abraham, A. (2015). Stock market forecasting using LASSO linear regression model. In *Afro-European Conference for Industrial Advancement: Proceedings of the First International Afro-European Conference for Industrial Advancement AECIA 2014* (pp. 371-381). Springer International Publishing.

Safari, F., Safari, N., & Montazer, G. A. (2016). Customer lifetime value determination based on RFM model. Marketing Intelligence & Planning, 34(4), 446-461.

Sawant, V. S. (2022). *Prediction of Customer Lifetime Value and Fraud Detection in BFSI using Machine Learning* (Doctoral dissertation, Dublin, National College of Ireland).

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008, October). Metrics for evaluating performance of prognostic techniques. In *2008 international conference on prognostics and health management* (pp. 1-17). IEEE.

Schmittlein, D. C., Morrison, D. G., & Colombo, R. (1987). Counting your customers: Who-are they and what will they do next?. *Management science*, *33*(1), 1-24.

Schrider, D. R., & Kern, A. D. (2018). Supervised machine learning for population genetics: a new paradigm. Trends in Genetics, 34(4), 301-312.

Shah, K., Patel, H., Sanghvi, D., & Shah, M. (2020). A comparative analysis of logistic regression, random forest and KNN models for the text classification. *Augmented Human Research*, *5*, 1-16.

Shih, Y. Y., & Liu, C. Y. (2003). A method for customer lifetime value ranking—Combining the analytic hierarchy process and clustering analysis. *Journal of Database Marketing & Customer Strategy Management*, *11*, 159-172.

Singh, S. S., Borle, S., & Jain, D. C. (2009). A generalized framework for estimating customer lifetime value when customer lifetimes are not observed. *Qme*, *7*, 181-205.

Sugo Music Group. (2016). IFPI Global Music Report 2016. *Sugo Music Group*. https://sugomusic.com/ifpi-global-music-report-2016/

Susic, P. (2023). 40+ Fascinating Music Streaming Statistics (2023). HeadphonesAddict. https://headphonesaddict.com/music-streaming-statistics/

Tapper, T. (2022). Using machine learning to predict customer lifetime value of players in a freemium mobile game: Effect of seasonal features.

Thach, N. N., Anh, L. H., & Khai, H. N. (2021). Applying lasso linear regression model in forecasting Ho Chi Minh City’s public investment. *Data Science for Financial Econometrics*, 245-253.

Therneau, T., Atkinson, B., & Ripley, B. (2022). *Recursive Partitioning and Regression Trees [R package rpart version 4.1.19]*. CRAN. https://cran.r-project.org/web/packages/rpart/index.html

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.

Tukel, O. I., & Dixit, A. (2013). Application of customer lifetime value model in make-to-order manufacturing. Journal of Business & Industrial Marketing.

Tyagi, A., Singh, V. P., & Gore, M. M. (2021). Improved detection of coronary artery disease using DT-RFE based feature selection and ensemble learning. In *International Conference on Advanced*

Network Technologies and Intelligent Computing (pp. 425-440). Cham: Springer International Publishing.

Venables, W. N., Ripley, B. D., Venables, W. N., & Ripley, B. D. (1997). Robust statistics. *Modern Applied Statistics With S-PLUS*, 247-266.

Venkatakrishna, M. R., Mishra, M. P., & Tiwari, M. S. P. (2021). Customer Lifetime Value Prediction and Segmentation using Machine Learning. *Int. J. Res. Eng. Sci*, *9*, 36-48.

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research*, *30*(1), 79-82.

Wlömert, N., & Papies, D. (2016). On-demand streaming services and music industry revenues—Insights from Spotify's market entry. International Journal of Research in Marketing, 33(2), 314-327.

Wong, T. T., & Yeh, P. Y. (2019). Reliable accuracy estimates from k-fold cross validation. IEEE Transactions on Knowledge and Data Engineering, 32(8), 1586-1594.

Wright, M. N., Wager, S., & Probst, P. (2023). *CRAN - Package ranger*. CRAN. https://cran.r-project.org/web/packages/ranger/index.html

Wright, S. (1921). Systems of mating. I. The biometric relations between parent and offspring. *Genetics*, *6*(2), 111.

WSDM. (2018). *WSDM - KKBox's Churn Prediction Challenge | Kaggle*. https://www.kaggle.com/competitions/kkbox-churn-prediction-challenge/overview

Yang, L., Liu, S., Tsoka, S., & Papageorgiou, L. G. (2017). A regression tree approach using mathematical programming. Expert Systems with Applications, 78, 347-357.

Zhang, W., Wu, C., Li, Y., Wang, L., & Samui, P. (2021). Assessment of pile drivability using random forest regression and multivariate adaptive regression splines. *Georisk: Assessment and Management of Risk for Engineered Systems and Geohazards*, *15*(1), 27-40.

Zhao, Q., & Hastie, T. (2021). Causal interpretations of black-box models. *Journal of Business & Economic Statistics*, *39*(1), 272-281.

Zhou, Q., Hong, W., Shao, G., & Cai, W. (2009). A new SVM-RFE approach towards ranking problem. In *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems* (Vol. 4, pp. 270-273). IEEE.

9 Appendix

9.1 Predicting customer lifetime value continued

This section explains the RFM model, the Pareto/NBD model, the BG/NBD model, and the Markov chain model in more detail, because these models are explained shortly in the literature review, as they are not being used in this research.

9.1.1 The RFM model explained in more detail

Recency, purchase frequency and monetary value are well known metrics used in marketing (Burelli, 2019). Together, these metrics are known as RFM and are often used to predict customer behavior (Gupta et al., 2006). Hughes (2005) proposed a method to estimate customer quality with RFM, where the customers of a company are divided into five quantiles for each variable, resulting in 5x5x5 groups. These groups are used to assign scores to customers to target them with tailored offers (Hughes, 2005).

Shih and Liu (2003) were inspired by Hughes (2005) and came up with a method based that uses RFM and CLV clustering to rank customers based on profitability. They explained that the first step relies on experts that need to identify the relative importance of the RFM variables using analytical hierarchical processing. After that, the customers are clustered based on RFM, and the clusters are scored using a weighted sum of the three normalized features (Shih and Liu, 2003).

Burelli (2019) stated that both methods can predict numerical values for CLV, but rather to rank the customers based on profitability. Additionally, he noted that these methods do not consider that customer behavior in the past is often the result of company actions in the past. Fader et al. (2005A) further emphasized that both methods are only able to predict customer behavior for one future period.

9.1.2 The Pareto/NBD and BG/NBD model explained in more detail

An alternative method to predict CLV is the pareto/NBD model introduced by Schmittlein et al. (1987). The model tries to the number of future purchases of customers based on recency, frequency, and customer lifetime (Glady et al., 2009B). It does this by using a Pareto distribution of the second kind and a negative binomial distribution (Burelli, 2019). Burelli explained that the Pareto distribution is controlled by the parameters s and β, while the negative binomial distribution is controlled by r and α. Parameter s represents the variation in customer lifetimes, β represents the average duration of a customer's lifetime, r represents the variability in the purchase frequencies of a customer and α represents the average purchase frequency (Schmittlein et al., 1987). Schmittlein et al. explained that these parameters can be estimated from past customer behavior by using the maximum likelihood or by fitting observed moments. With these parameters, the model can predict the number of future purchases for each customer based on recency, frequency, and customer lifetime (Burelli, 2019).

The computational complexity is one of the drawbacks of the Pareto/NBD model (Fader et al., 2005B). to address this issue, the modified BG/NBD model was proposed by Fader et al. (2005B), which can be implemented more efficiently (Burelli, 2019). Feder et al. stated that customer activity is modeled based on parameters p and q in the BG/NBD model, where p represents the probability of a customer making a purchase within a specific time, while q represents the probability of the customer becoming inactive after making a purchase. Both models, as explained by Burelli (2019), can predict the future number of purchases for a customer and estimate the number of active customers at a given point in time. However, he explained that both models cannot model the value of each purchase, making them unable to predict customer lifetime directly.

Reinartz & Kumar (2003) utilized the Pareto/NBD model to estimate the number of time periods in which a customer is expected to make a purchase. They transformed a continuous variable that represents probability whether a customer is active, into a binary variable that determines whether a customer is active or not at a specific time based on a probability threshold. This variable makes it possible to identify when a customer will churn and, when combined with the start date of the customer relationship, it can also estimate the expected customer lifetime (Reinartz & Kumar, 2003). The lifetime, expressed as n periods, can be used to calculate the customer lifetime value using the basic CLV calculation method (Burelli, 2019).

According to Burelli (2019), one of the main advantages of both models is that they only need historical transactional data, making them easily applicable in various contexts. However, the author explained that this advantage also poses a drawback, as the models may overlook important information, resulting in suboptimal models. To address this limitation, Singh et al. (2009) proposed an estimation framework that allows the inclusion of multiple statistical distributions and covariates such as age and gender.

9.1.3 The Markov Chain model explained in more detail

According to Pfeifer & Carraway (2000), an alternative method to predict CLV by using a Markov Chain Model (MCM) to model the customer relationship. MCMs are mathematical models that describe random processes (Ching & Ng, 2006). Ching & Ng explained that a process is represented by a set of states, and transitions between these states are determined by probabilities. Moreover, they explained that the transitioning of each state to another state has its own associated probability, called a transition probability, and the probabilities are often shown in a square transition matrix. An important characteristic of a MCM is that the future behavior of the process only depends on its current state and is unaffected by previous states (Ching & Ng, 2006).

Pfeifer & Carraway (2000) applied the MCM in a way where the states represented different relationship conditions between customers and the company, and where the transition probabilities between the states represented the probability of a customer moving from one condition to another, for example to churn or to make a purchase.

According to Burelli (2019), calculating the transition probabilities can be done in three steps. Firstly, a that a transition matrix should be created where the value of each cell should be equal to zero. Secondly, for every customer that transitions from state i to state j, the corresponding cell (ij) should be increased. Finally, each row of the matrix should be normalized to a range between 0 and 1 using a min-max normalization. Burelli (2019) suggested that with this final matrix, CLV can be calculated for every possible state.

9.2 Variable descriptions

Table A1 Variable descriptions and formats of the variables in the raw data sets

Variable	Description	Format	Members	Train	Transactions
msno	user id	character	x	x	x
city	The city that the user lives in indicated by a number instead of an actual city name	integer	x		
bd	The age of the user	integer	x		
gender	The gender of the customer (male; unknown; female)	character	x		
registered_via	Registration method indicated by a number instead of an actual method	integer	x		
registration_ init_time	The date when a user registered	integer	x		
expiration_date	The expiration date of a customer when members.csv is extracted. It does not represent churn.	integer	x		
is_churn	A binary variable indicating whether a user did not continue the subscription within 30 days after the expiration date	integer		x	
payment_ method_id	Payment method indicated by a number instead of an actual method	integer			x
payment_ plan_days	Length of the membership plan in days	integer			x
plan_list_price	Expected fee to be paid in New Taiwan Dollar (NTD)	integer			x
actual_ amount_paid	Actual fee paid in NTD	integer			x
is_auto_renew	A binary variable indicating whether a user's subscription is auto renewed	integer			x
transaction_date	The date of a transaction	integer			x
membership_ expire_date	The date when a membership expires	integer			x
is_cancel	A binary variable indicating whether the user canceled the membership in a during a certain transaction or not	integer			x

Table A1 Variable descriptions and formats of the variables in the final aggregated data set.

Variable	Description	Format
user_ID	User ID	integer
gender	The gender of the customer (male; female)	Factor w/ 2 levels
bd	The age of the user	integer
city	The city that the user lives in indicated by a number	Factor w/ 21 levels
registered_via	Registration method indicated by a number instead of an actual method	Factor w/ 5 levels
is_churn	whether a user did or did not continue the subscription within 30 days after the expiration date	Factor w/ 2 levels
frequency	The number of payments made by the user over the entire period	integer
average_amount_paid	The average fee price paid by the user	numeric
recency_months	The number of months between the last transaction made by the user and the end of the period	numeric
number_months_from_start	The number of months between the beginning of the period and the first transaction made by the user	numeric
CLV	The customer lifetime value	numeric

9.3 Additional figures from the analysis

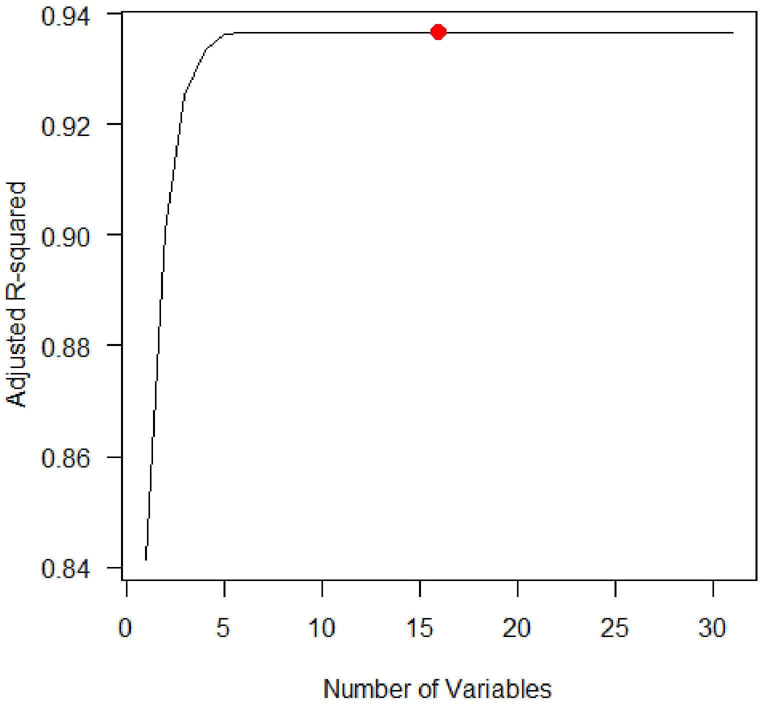

Figure A1 Adjusted R-squared plotted against the number of variables included in the best subset

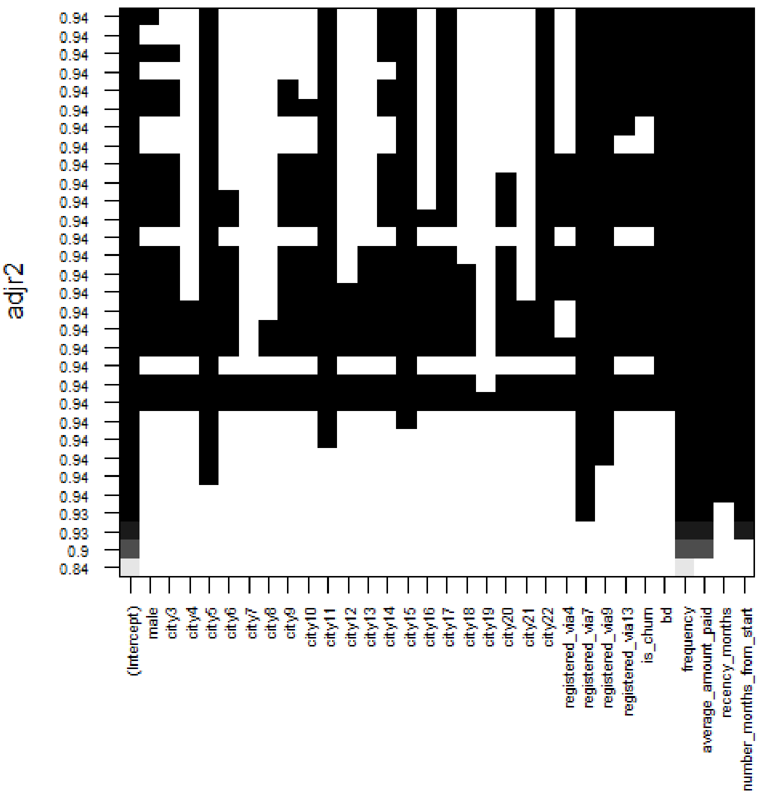

Figure A2 Adjusted R-squared plotted against all subsets

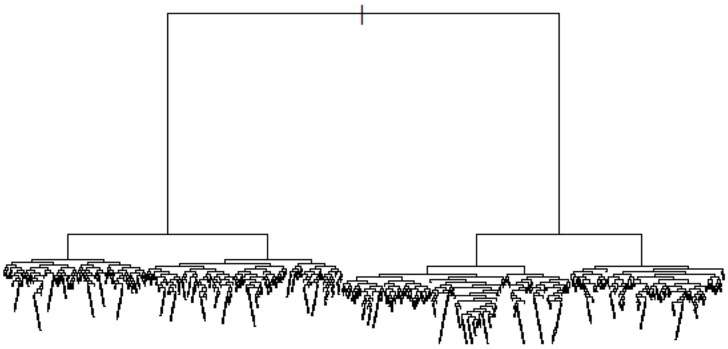

Figure A3 Fully grown RT plotted

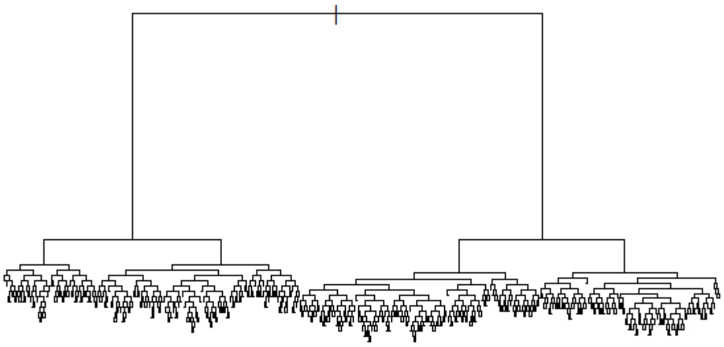

Figure A4 Pruned RT plotted

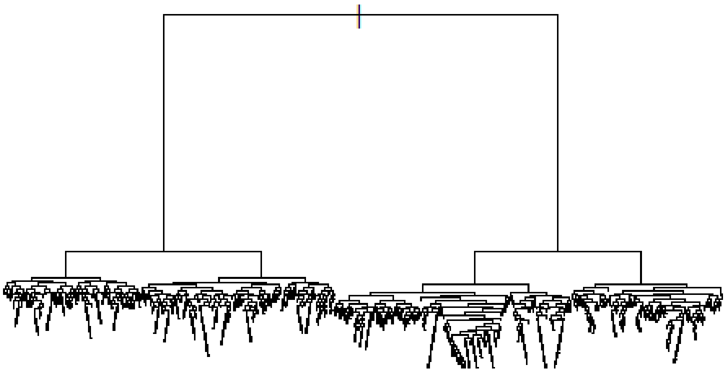

Figure A5 Fully grown RT plotted after performing BSS

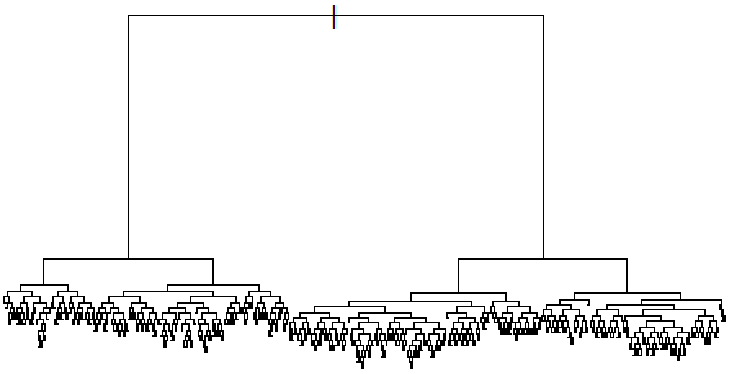

Figure A6 Pruned RT plotted after performing BSS

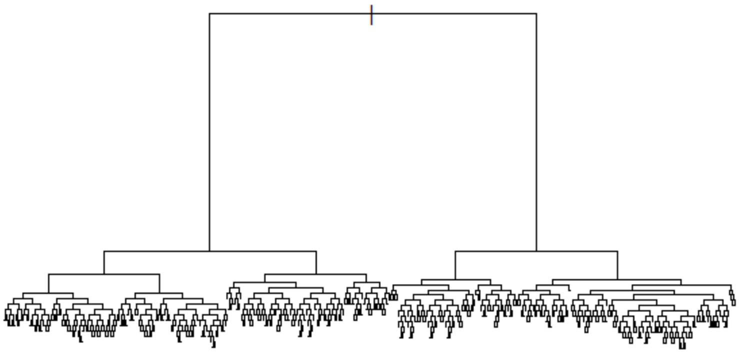

Figure A7 Fully grown RT plotted after performing RFE

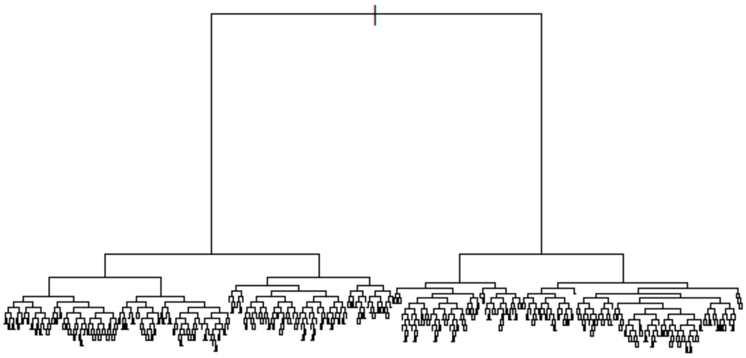

Figure A8 Pruned RT plotted after performing RFE