Uniwersytet Warszawski
Wydział Nauk Ekonomicznych

Jarosław Kupisz
Nr albumu: P-18826

Client Churn Prediction with Data Mining. Literature Review and SaaS B2B Case Study

Studia podyplomowe
„Data Science w zastosowaniach biznesowych. Warsztaty z wykorzystaniem programu R"

Praca dyplomowa
wykonana pod kierunkiem
dr Piotra Wójcika
Zakład Finansów Ilościowych

Warszawa, listopad 2018

Oświadczenie kierującego pracą

Oświadczam, że niniejsza praca została przygotowana pod moim kierunkiem i stwierdzam, że spełniła ona warunki do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data Podpis kierującego pracą

Statement of the Supervisor on Submission of the Thesis

I hereby certify that the thesis submitted has been prepared under my supervision and I declare that it satisfies the requirements of submission in the proceedings for the award of a degree.

Date Signature of the Supervisor:

Oświadczenie autora pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data Podpis autora (autorów) pracy

Statement of the Author on Submission of the Thesis

Aware of legal liability I certify that the thesis submitted has been prepared by myself and does not include information gathered contrary to the law.

I also declare that the thesis submitted has not been the subject of proceedings resulting in the award of a university degree.

Furthermore I certify that the submitted version of the thesis is identical with its attached electronic version.

Date Signature of the Author(s) of the thesis

Abstract

Client churn prediction became an increasingly important classification problem, as an offering-related competitive advantage is yielding its place to Customer Relationship Management for securing long-term growth. This study presents recent findings of churn prediction literature ordered by model creation phases and their application in SaaS B2B experiment. Results indicate that ways of data preprocessing devised by cited scholars offer improvement of up to 17% over standard approaches, validate novel Maximum Profit criterion (outputting direct profits from a model) as an evaluation metric, and highlight Logit Leaf Model featuring enchanted interpretability as a viable algorithm for churn prediction.

Table of Contents

Introduction

Churn prediction is becoming an increasingly important classification problem in a business setting. Data mining literature cover this topic extensively, very frequently on examples of financial and telecommunication sectors[1]. Companies on these markets, especially in developed countries, due to the fundamental nature of needs that they are satisfying, deal with customer bases covering nearly whole economically active population[2]. Although modern business strategy scholars advise focusing company's efforts on non-consumers[3], these companies have little breathing room in finding new sources of revenue, when it is not uncommon to meet people with multiple smartphone devices and owning a plethora of financial products. Research focus on these sectors is also enabled by huge data warehouses allowing for fairly easy retrieval of the multi-year history of behavioral and transactional data accompanied by many demographical attributes[4]. However, client churn modeling can be used outside of the financial and banking industries. It is a general classification problem in today's economy, where civilizational advancements can render many types of competitive advantage obsolete. Although not every market category is as saturated, has profitability vulnerable to high attrition rates and offers an abundance of customer-related data, retaining existing customer with incentives is simply cheaper than acquiring and converting a new one into a loyal client[5]. Moreover, with easier access to advanced technologies, the axis of competitive advantage is being shifted towards constructing mutually beneficial long-term relationships with clients. ***Customer Relationship Management*** (CRM) becomes then an essential function of an enterprise, with client churn prediction as one of the main tools serving to maximize its return on investment.

The aim of this study is to present to the reader all necessary terminology and recent findings of churn prediction literature, which would prepare him to tackle this classification problem in

[1] Verbeke Wouter, Dejaeger Karel, Martens David, Hur Joon, Baesens Bart (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research*, 218, 1, p. 212.

[2] Larivière Bart, Van den Poel Dirk (2005). Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications*, 29, 2, p. 475.

[3] Kim, C.W., Mauborgne, R. (2005). Blue Ocean Strategy: From Theory to Practice. *California Management Review*, 47, 3, p. 105-107.

[4] Chen Zhen-Yu, Fan Zhi-Ping, Sun Minghe (2012). A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *European Journal of Operational Research*, 223, 2, p. 466.

[5] Larivière Bart, Van den Poel Dirk (2005). Predicting customer retention and profitability by using random forests and regression forests techniques. Expert Systems with Applications, 29, 2, p. 483.

their companies. Not only this work presents the theoretical underpinnings of churn prediction but offers an example of practical application of these principles in currently modern software as a service (SaaS) B2B setting. In order to do so, the first chapter links the managerial world and mathematical methods of data mining (popularly known as *data science*), by listing main tools, phases, and objectives of CRM, and how data modeling can intertwine with them. CRM imposes many limitations on classical classification methods and they are also indicated at the end of this chapter.

Chapter two starts by describing a recommended process for creating data-mining models in for-profit enterprises (CRISP-DM). Afterward, each phase important for constructing churn prediction classifier is investigated in detail, bearing in mind implications of CRM paradigm. Firstly, during database construction, types of recommended predictors, temporal lagging of variables and their aggregations are discussed. Then, a detailed discussion on recommended data preprocessing by churn scholars is offered. Next, four benchmark classification algorithms (logistic regression, decision trees, random forests and support vector machines) are presented with the addition of logit leaf model – a specialized ensemble approach designed explicitly for churn prediction, motivated by the need of increased interpretability while maintaining the efficiency of benchmark methods. After a brief discussion of feature selection, sampling and cross-validation methods, the second chapter concludes with a discussion on churn prediction model evaluation metrics. Critique of standard approaches, such as AUC and confusion-matrix based measures, is followed by a presentation of top-decile lift and novel maximum profit criterion.

Chapter three puts methods described previously to practical use. An experiment is designed and conducted on a dataset provided by a SaaS B2B company. Each of 5 algorithms is trained on 3 flavors of the dataset, each preprocessed differently. Results are analyzed under three hypothesis, designed in a way to construct useful recommendations for practitioners willing to use described methods. Firstly, the superiority of data preparation methods advised by churn prediction researches is examined. Secondly, the viability of Maximum Profit Criterion, a novel evaluation metric maximizing returns from a retention campaign and optimizing fraction of customers to target, is considered. Finally, the logit leaf model is juxtaposed with other types of classifiers to justify its production use. To increase practical implications of the experiment, it was conducted fully with very popular R programming language and publicly available modules. The study concludes with pointing areas for future research.

Chapter 1. Customer Churn Prediction under CRM Paradigm

This chapter presents a place of managing customer churn in the customer relationship management strategy of any enterprise and defines key data mining concepts related to it.

1.1 Customer Relationship Management (CRM)

E.W.T. Ngai et al. (2009) define ***Customer Relationship Management*** (CRM) as a set of processes and enabling systems, which are supporting a business strategy to build long-term, profitable relationships with specific customers[6]. Unfortunately, there is no unique generally accepted definition, although this one incorporates most recurring elements among many[7]. The "processes and enabling systems" can be divided into ***operational*** and ***analytical***. Division implications can be easily derived from their names: the former aims to automatize and perform cost-efficiently actions needed to put CRM strategy into motion, while the latter tries to use company's customer-related data to shape it and optimize relationships with individual customers. This optimization means that company's resource allocation process becomes oriented towards specific customers, who are classified as promising in terms of future mutual benefits, instead of trying to wastefully and somewhat blindly "concentrate" efforts on the whole customer base. Due to the nature of this work only concepts related to analytical CRM systems and processes will be discussed.

CRM is composed of 4 consecutive steps, which contain all phases of customer lifetime under CRM paradigm:

1) Customer Identification
2) Customer Attraction
3) Customer Retention
4) Customer Development

These steps are geared towards enabling long-term customer relationship by gradually analyzing customer's needs to achieve a deep understanding of them and making sure that customer won't flee to competition and/or decrease their commitment. The main goal of the

[6] Ngai E.W.T., Xiu Li, Chau D.C.K (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications* 36, 2, 2, p. 2592.
[7] Ibidem, p. 2593.

first of them is to identify a population of potentially profitable customers via analysis of population's, segments' and individual customer's characteristics. ***Customer identification*** can be performed by simply trying to match customers with company's offering by reasoning, using more sophisticated data mining methods, as well as analyzing direct competitor's clientele, to uncover reasons of losing them. ***Customer attraction*** descends from population and segment level down to preparing direct marketing offers to individual customers who were deemed as attractive. These direct marketing tools can be but are not limited to: email, direct mail, phone calls or new-joiner incentives delivered to potentials purchasers via any communication channel. ***Retaining a customer*** is considered to be the most important phase, as it directly supports maintaining long-term, economical value creating relationships between both parties. Activities within this step can be divided into one-to-one marketing, loyalty programs, and complaints management. First of all, a customer needs to be satisfied with the purchased offering, therefore complaints management seeks to create a feedback loop, which allows ensuring that client's return on investment achieved satisfactory level. Common across customers churn-preventing incentives can be delivered via loyalty programs, which consist of promotional campaigns and supporting activities encouraging customers to stay longer. One-to-one marketing is used in cases which require more of company's attention and cannot be addressed by global campaigns. Reasons for offering individualized inducements most often is economical attractiveness of a relationship, however strategical considerations are often taken into account. Lastly, when the relation is beneficial, customer development tools are used to maximize transaction intensity and value, ideally together with profitability. Customer lifetime value is a critical metric at this stage, which allows managers to order clients by past and/or expected income from a relationship. Up/cross-selling then can be used to increase the purchase of complementary and associated services and products. Another typical framework used at this point is market basket analysis, which exposes the customer's pattern of successively buying products, nudging other customers to follow same purchasing habits[8].

1.2 Data Mining and its Implications in CRM

Data mining methods can be used in any of described CRM stages in order to, for example, segment customers based on their behavior, list customers who should be eligible for loyalty

[8] Ibidem, p. 2594-2595.

incentives or perform a regression analysis to understand which features have the greatest impact on their sensitivity to downgrade. Virtually any marketing tool listed in the previous paragraph can be supported by data mining models, therefore data mining definition needs to be introduced at this point. Generally, ***data mining*** can be seen as a process using statistical, mathematical, artificial intelligence (AI) and machine learning (ML) methods to extract and identify useful information and gain knowledge from large databases[9]. It belongs to a broader set of frameworks called ***Knowledge Discovery in Databases*** (KDD), aiming at the extraction of useful information from raw data in large databases[10]. In the context of CRM, one can easily project knowledge underlined in these definitions onto an understanding of customers' needs over their lifetime, and large databases onto data warehouses of companies containing customer attributes and transactional history. Although the idea of gathering client-level data predates methods and concepts discussed later in this study[11], advancements in the availability of cheap and powerful computing resources within last decade, allowed data mining methods to truly take off in CRM context. Their usage by firms is considered a growing global trend. Another of their enabling features is the ability to search for interesting patterns among many covariates coming from multiple terabyte-scale data sources, giving businesses the ability to generate insights which were prohibitively costly to attain in the past[12].

Data mining tools, broadly speaking, achieve their goal of generating useful information via building model generalizing dependencies between variables in the target dataset. These models can belong to one or several of the following categories of data modeling:

1) **Association**: Association models establish a relationship between records for one dimension instance, in the context of this study – a customer. CRM specific applications of this kind of modeling would be market basket analysis and cross-selling programs.

2) **Classification:** Classification algorithms are the cornerstone of data mining, as they aim to predict class values based on provided variables. In CRM they are used to predict future customer behaviors and states, including churn prediction among many.

[9] Ibidem, p. 2593.
[10] Verbeke W., Dejaeger K., Martens D., Hur J., Baesens B. (2012). New insights into…, op. cit., p. 212.
[11] Kimball Ralph, Ross Margy (2016). *The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence*. Indianapolis: Wiley, p. 37-47.
[12] Coussement Kristof, Van den Poel Dirk (2008). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications*, 34, 1, p. 315.

Virtually any classification algorithm can be applied (even simple if-then rules) and detailed discussion on them is presented in the second chapter.

3) **Clustering:** Clustering aims to divide a heterogeneous sample into more homogeneous clusters. Although similar at first glance with classification, the main difference is that possible clusters are unknown before a run of the algorithm and their number is often arbitrary chosen by the user. Clustering methods fall into the unsupervised learning family of algorithms. K-Nearest Neighbors and discrimination analysis are often used by CRM practitioners.

4) **Forecasting:** Again this category has similarities with classification, however forecasting typically deals with continuous outcomes, such as demand for a product. These methods try to predict future values at chosen point in the future based on past dependencies between variables or evolution of a variable in time. Time-series methods constitute a popular example. This category of modeling tools can also be used to forecast a class of a sample at certain point in time. In the context of client management, survival analysis can be used to predict their spending category in the next period.

5) **Regression:** Regression algorithms are classical statistical tools which map an observation's variables to a real value, then used typically for prediction. It often involves finding a curve which minimizes the least squared error, modeling relationships (how dependent variable will change if one of independent variable varies) and testing hypothesis whether a variable is informative. Linear regression is still widely used, not only in the CRM context. Related methods which map output to a limited range of values, such as logistic regression, can be also used for classification tasks. As will be discussed in chapter 2, logistic regression is still considered as a legitimate benchmark for modern and far more complex methods of churn prediction.

6) **Sequence discovery:** Sequence discovery's goal is to model states of processes generating a sequence of steps. These models are then frequently used for deviation analysis, for example finding customers who have not performed an action which was expected of them by that model. Various statistical approaches can fulfill this task.

7) **Visualization:** Data visualization is not a data modeling method on its own, however, it is usually used in conjunction with all other categories. As visual perception can assimilate much more information than simply reading[13], using graphical objects in a two-dimensional coordinate space can be used to highlight complex patterns in an easy to grasp way. Another virtue of visualization is easier communication of model results, as persons responsible for CRM do not always have a quantitative background.

As shown earlier, CRM does not have a single definition, therefore its functions, methods, processes together with a length of customer lifecycle vary by industry, with the size of a company and can be influenced by many other factors. The same applies to data mining methods used to support analytical CRM, as recommendations in the literature vary often from paper to paper, being contradictory at times[14]. This flexibility is well depicted in figure 1, as any of the discussed data mining classes can be "picked-up" to support marketing tools used in all four phases of CRM.

[13] Tufte Edward R (2001). *The Visual Display of Quantitative Information*. Connecticut: Graphics Press, p. 76-77.

[14] Many cited papers claim superiority of one algorithm over another consecutively. Detailed discussion on this in: Verbeke W., Dejaeger K., Martens D., Hur J., Baesens B. (2012). New insights into…, op. cit.

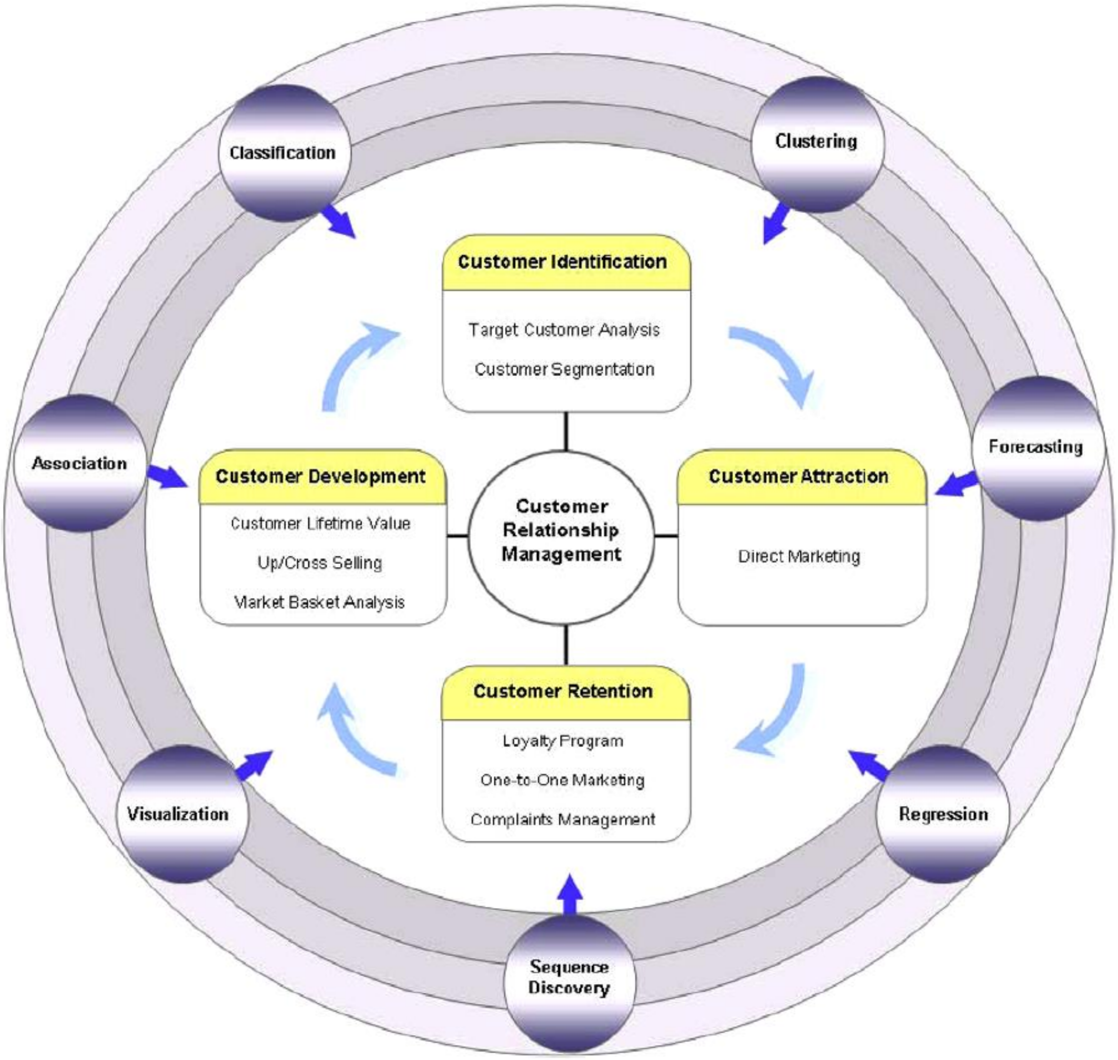


Figure 1. Phases of CRM and types of data modeling. Source: Ngai E.W.T., Xiu Li, Chau D.C.K (2009). Application of data mining…, op. cit., p. 2594.

Nonetheless, client churn prediction is a problem where data mining methods have to be applied with consideration. Solving economic problems, such as efficient CRM management, is subject to constraints of creating economic value. This fact, together with CRM definition pose certain difficulties, towards which generic data mining approaches are not always well suited. For example, although most of the classification algorithms output probability of an observation belonging to a certain class, selecting a cutoff point to determine which customers will participate in churn preventing campaign is of utmost importance. Selecting too many will undermine the profitability of such an enterprise by not breaking even on fixed costs of preparing customer incentives, while selecting too few might result in big alternative costs[15]. Additionally, as customer churn prediction applies often to contractual settings, where dataset granularity descends to a contract level, for companies operating for many years much more non-churning data points are observed. Applying simple accuracy metric would then result in over 90% when simply predicting all contracts as extending. Therefore not only model evaluation metrics have to be attentively selected, but observation sampling for training data seems of much greater importance than in other classification settings. On top of that, the applied model has to be allied with business knowledge and possibly expand it with novel reasoning. "Black-box" algorithms are then less desirable, as they do not augment marketing intelligence outside of data mining team. These and other impacts of CRM customer retention process on data mining modeling, together with methods addressing them proposed by scholars in this field, are subjects of the following chapter.

[15] Verbeke W., Dejaeger K., Martens D., Hur J., Baesens B. (2012). New insights into…, op. cit., p. 225-229.

Chapter 2. Overview of Customer Churn Prediction Data Mining Approaches Based

As chapter 1 laid the foundations for CRM Customer Retention environment in which data mining models should produce profitable results, this chapter offers a discussion of methods which should be befitting for bringing desired outcomes. This part of the study opens with a discussion on the process of creating a data mining model for business problems solving, then it elaborates on them using data mining methods by presenting an apparatus used in it. Standard approaches are juxtaposed with client churn predictions methods proposed by scholars working actively in this field during the last dozen years. Chapter 2 concludes with a selection of approaches, which will be used to verify the research hypothesis in chapter's 3 experiment.

2.1 Data Mining Process in Business Setting

2.1.1 CRISP-DM Data Mining Process

Cross-Industry Standard Process for Data Mining (CRISP-DM) is a high-level process template which can be used for business problems solving with data mining. It allows imposing a structure of distinct phases with feedback-loops between them for greater objectivity, better progress tracking and iterative gain of knowledge by employees[16].

[16] Provost Foster, Fawcett Tom (2013). *Data Science for Business*. Sebastopol, CA: O'Reilly Media, p. 27.

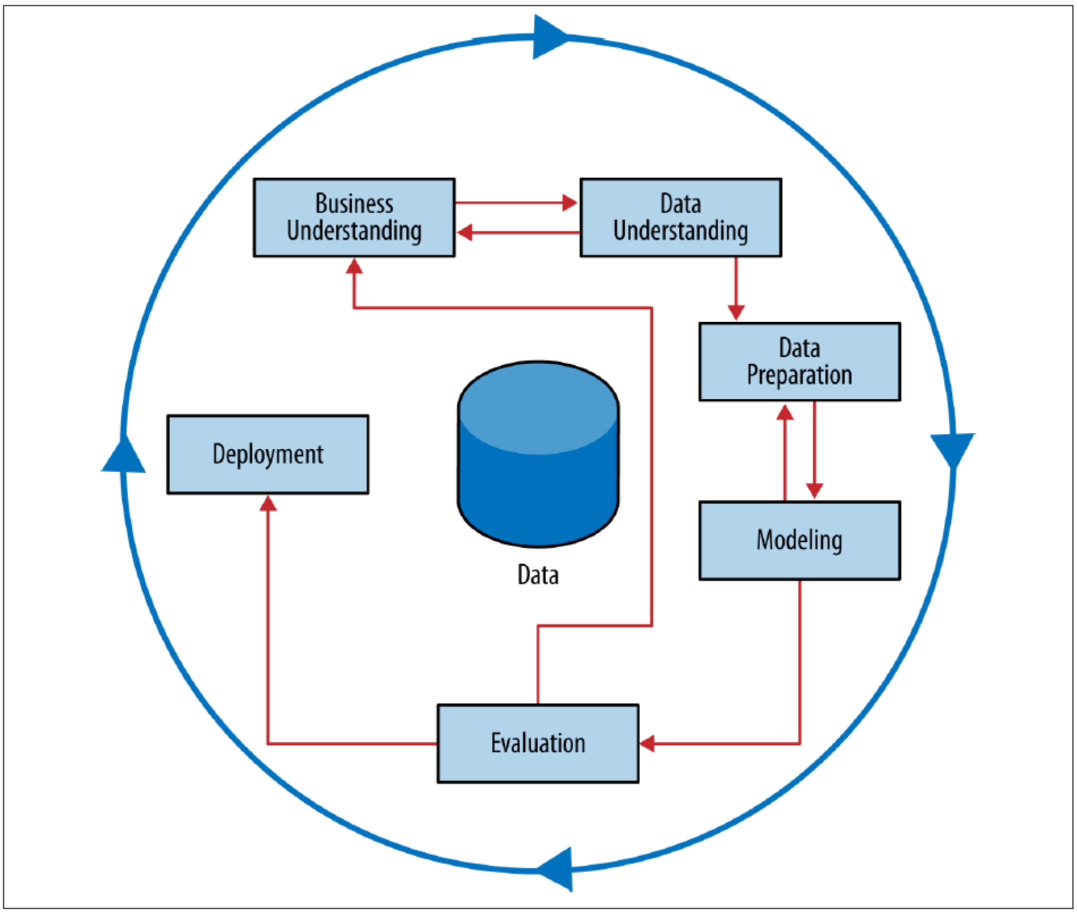


Figure 2. CRISP-DM process diagram. Source: Provost Foster, Fawcett Tom (2013). Data Science…, op. cit., p. 27.

First, and arguably the most important stage is ***business understanding***. During it, business analysts try to translate a business problem into a data mining one, which can be tackled by types of modeling discussed in chapter 1. This endeavor requires a great deal of creativity, organization and market knowledge, as two quite different sciences have to collide to produce economically viable solutions. However no matter how creative the analysts are, what can be achieved with data mining approach is limited by the data available and its quality. At this point, the project team has to weigh the costs and benefits of acquiring new data and/or improving the quality of existing databases. This trade-off ultimately limits what can be achieved and bi-directional arrows between the first two phases illustrate constant reformulation of the business problem. When the goal is settled and seems achievable, one enters the ***data preparation*** stage where a dataset is transformed into a tidy format[17], typically

[17] Features of tidy data are discussed in: Wickham Hadley (2014). Tidy Data. *Journal of Statistical Software*, 59, 10, p. 1-23.

involving tabularizing unstructured data and preparing features necessary for prediction task. ***Modeling*** is where data mining models are created and initially evaluates to select best-performing algorithms and reveal regularities in the dataset. Evaluation stage is not simply assessing the power of a model with performance metrics. Its purpose is to confirm that models perform correctly and efficiently in a real business scenario, but also whether all stakeholders support the way the solution it tackling the original business problem. If no irregularities are found model is then ***deployed*** into some kind of organization's information technology system or incorporated into a business process, such as CRM. It is not uncommon to see modeling teams at this point going back to square one (business understanding), due to lack of the desired impact on business. Even though this might be seen as a failure, frequent feedback loop between stages ensure that the organization gains new knowledge from the process at every stage[18].

Presentation of CRISP-DM model was necessary to form a bridge between the managerial problem of churn prediction and numerical approach of data mining, allowing to impose structure and order of reviewing churn prediction methods. However, it is very high-level and involves stages which won't be discussed in this study. Verbeke et al.[19] (2012) in their overview of churn prediction methods, propose a process which can be seen as a close-up of the CRISP-DM subset. It won't be formally introduced here like its generalized counterpart, nevertheless following sections will be structured as seen in the first two steps of figure 3.

[18] Provost Foster, Fawcett Tom (2013). *Data Science*…, op. cit., p.27-33.

[19] Verbeke W., Dejaeger K., Martens D., Hur J., Baesens B. (2012). New insights into…, op. cit., p.212-213

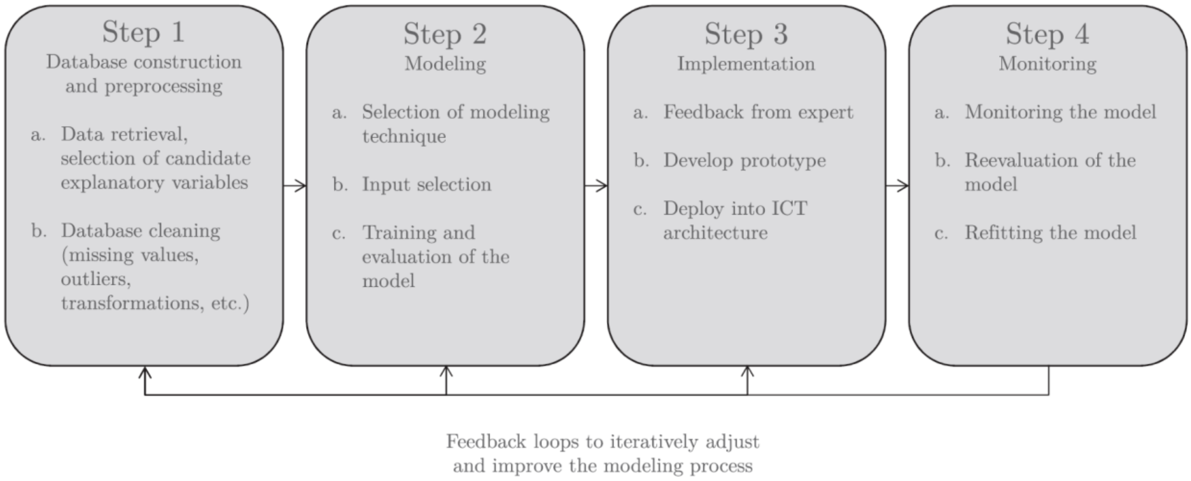

Step 1
Database construction and preprocessing
a. Data retrieval, selection of candidate explanatory variables
b. Database cleaning (missing values, outliers, transformations, etc.)
Step 2
Modeling
a. Selection of modeling technique
b. Input selection
c. Training and evaluation of the model
Step 3
Implementation
a. Feedback from expert
b. Develop prototype
c. Deploy into ICT architecture
Step 4
Monitoring
a. Monitoring the model
b. Reevaluation of the model
c. Refitting the model
Feedback loops to iteratively adjust and improve the modeling process

Figure 3. The process of the development of a customer churn prediction model. Source: Verbeke W., Dejaeger K., Martens D., Hur J., Baesens B. (2012). New insights into…, op. cit., p.213

2.2 Database Construction and Preprocessing

Although CRM context imposes several particularities on data mining models as already discussed, at its core each classification problems requires a training dataset consisting of *n* 2-tuples {(x_1, y_1), …, (x_n, y_n)}, where x_n is a one-dimensional vector composed of D predictive features X. Therefore X = {$X_{1,}$ $X_{2, \ldots,}$ X_D}, where each X contains n values, one for each (in setting of this study) customer or contract. *Y* is a vector of the dependent binary variable containing n observations, while $y_i \in$ {0, 1}. It indicates whether a sample is a churner (1) or a contract extending customer (0). ***Training dataset*** is used to optimize parameters of a generalization function which tries to spot regularities in n x_i vectors to discriminate churning observations from loyal ones. Then ***testing dataset***, which has exactly the same form as training dataset (with identical D vectors features X), with samples unseen by the model is used to evaluate generalization power of an algorithm[20]. Although details of training procedure are discussed later, this section tries to reply to the question of how to construct such a dataset in churn prediction setting, what kind of variable a practitioner should be most keen towards, and what kind of initial transformations should be applied to maximize model's performance.

[20] De Bock Koen W., Van den Poel Dirk (2012). Reconciling performance and interpretability in customer churn prediction using ensemble learning based on generalized additive models. *Expert Systems with Applications*, 39, 8, p. 6817.

2.2.1 Data Retrieval and Selection of Candidate Explanatory Variables

In the context of churn prediction, data retrieval usually depends on how an organization stores its data. In organizations with mature Business Intelligence (BI) systems which deliver on the promise of providing conformed dimensions tables allowing for filtering of transactional tables, pulling a dataset together might be a matter of a simple database query. In less analytically mature organizations, scattered data sources pose a great challenge to acquiring the required training dataset in a timely manner, sometimes spurring an endless feedback loop between the first two stages of the CRISM-DM model[21]. Unfortunately, as each organization's circumstances are unique when it comes to data retrieval, therefore client churn prediction scientist do not offer any particular tool or advice. Nevertheless, Verbeke et al. (2012) study gives a general idea of what kind of variables to search for, if one is starting truly from scratch. Although their research was carried out in the telecommunications sector, the four broad types of variables they distinguish can be applied universally. First of all, metrics describing the ***degree of product/service usage*** offered by a company accounted for 40% of variables used by best performing techniques. Although intuitive, quantification of this fraction offers precise guidance. ***Financial* (**such as the value of a contract), ***marketing*** (eg. tier of the used solution, number of contacts with a company) and ***socio-demographic*** variables (customer's age, sex, size if institutional etc.) amount to the remaining 60%, while having even shares. This means that no type of predictor can be left out and has to be considered by a modeler[22]. Business domain experts are a valid source of creative feature engineering, which often decides the success of a modeling initiative.

Although the most powerful, behavioral features can become a double-edged sword. In order for an incentive campaign to bear fruits, a sufficient lag period between prediction moment and potential churn occurrence is needed. This logic arises from two reasons. First, a practical one – CRM team needs time to prepare incentives, distribute them and enter negotiations with customers. The second can also be seen as practical but from the customer's point of view. Often when someone has decided to switch providers, their usage increases drastically in order to reap all the benefits of an already paid contract. The question to be addressed by data mining team is whether a peak in usage variable is an indication of *potential churn* or one *that has already taken place*. This problem can be solved by ***lagging variables*** behind in time

[21] Kimball Ralph, Ross Margy (2016). *The Kimball Group… op.cit.,* p. 75–118.
[22] Verbeke W., Dejaeger K., Martens D., Hur J., Baesens B. (2012). New insights into… op. cit., p. 227

against churn dependent variable value. Window's length has to be decided on a case by case basis, but literature provides some guidelines. For example, Verbeke et al. (2012) recommend that advance of churn flag be at least one month, but not more than three[23]. Of course, a numerical optimization is always an option, yet practical realities of the length of marketing processes will ultimately decide possible range. Therefore lagging predictors cannot be performed without validation of a business expert[24].

Usage features entail yet another research problem. These predictors are often longitudinal in nature, and they have to be somehow summarized to serve as input to a classification algorithm. This aggregations often represent ***recency, frequency***, or translate ***usage into monetary value***. Recency can be represented as days from the last call, up to the prediction moment. To create frequency features positional statistics are used, with mean and median being by far the most popular, such as mean monthly minutes of usage. Same can be said for monetary predictors, with examples such as total and mean recurring charge. Zhen-Yu Chen et al. (2012) rightly notice that although this is a necessary step, a loss in resolution of data can be very costly. Positional statistics such as mean, although popular due to their broad understanding, are often used without checking the underlying distribution, misinforming the prediction algorithm. Although detailed explanation is out of the scope of this study, their line of research proposing usage of ensemble classifier composed of temporal features classifier and static predictors classifiers is noteworthy[25].

2.2.2 Data Cleansing

When a dataset with features accepted by common business knowledge, composed of a mix of static and aggregated temporal features was constructed, the next step is to perform ***data cleansing***. Again, not only there is no single universally accepted definition for this phase but it is also known under many names such as data preprocessing[26], data preparation[27], data

[23] Ibidem p. 227.

[24] Chen Zhen-Yu, Fan Zhi-Ping, Sun Minghe (2012). A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. European Journal of Operational Research, 223, 2, p. 465-466.

[25] Ibidem, p. 461-462.

[26] Vafeiadis T., Diamantaras K.I., Sarigiannidis G., Chatzisavvas K.Ch (2015). A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*, 55, p. 4

[27] Coussement Kristof, Lessmann Stefan, Verstraeten Geert (2017). A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry. *Decision Support Systems*, 95, p.29.

wrangling[28]. What is more, these names are often used interchangeably in the same article, therefore the same will be done in this work. Coussement et al. (2017) define this stage simply as: *a process that aims to convert independent (categorical and continuous) variables into a form appropriate for further analysis*. They also confirm its placement in data mining timeline after data collection, extraction, and variable construction. They also classify data preparation treatments methods into two categories, which are usually employed one after another: ***value transformation*** and ***value representation***[29].

Value transformation aims to reduce the dimensionality of discrete dependent variables and transform continuous predictors into discrete ones. This reduced input to the classification algorithm allows for faster convergence and increased scalability of a model, while not reducing significantly discriminative power. This is also the step during which outlier handling and missing values imputation strategies are decided. Value representation depends on algorithms used for prediction, as its goal is to alter predictors with their values transformed into a form suitable for input. The most common use case would be the inability of logistic regression to handle multi-category discrete variables and need to represent them as a set of binary flags.

2.2.2.1 Value Transformation Techniques

Value transformations methods are different for categorical and continuous variables. For discrete variables, the main goal is to reduce the number of individual categories from m to k where k < m and preferably by a large margin. However not always this step is necessary, especially when a predictor contains a small number of unique values, with its' distribution being relatively even. While not being really a method, ***no regrouping*** has to be used with consideration, preferably backed by input from business savvy persons. ***Remapping*** however not only creates new categories from original ones but does so while ensuring maximal homogeneity of a new group against target variable. Coussement et al. (2017) recommend using decision trees as they are tailor-made for this purpose. Applying a decision tree divides successively a predictor's space into a set of nodes, trying to choose splitting values in a way that maximizes the share of churners or non-churners in each terminal (i.e. last) node. To avoid "growing" a decision tree too big (as our goal is to reduce the number of unique

[28] Wikipedia (2018). *Data wrangling*. URL: https://en.wikipedia.org/wiki/Data_wrangling. Date of access: Aug. 16th 2018.

[29] Coussement Kristof, Lessmann Stefan, Verstraeten Geert (2017). A comparative analysis… op.cit., p.29

categories, not exceed it), a tree pruning strategy needs to be employed. Its goal is to find the smallest subtree which retains discriminative power on a similar level as a fully grown tree. At the end of a decision tree, all customers belonging to the same terminal node, get the same label in the new remapped feature space. Another advantage of using decision trees is that outliers are handled gracefully, and at the end of the process they simply belong to one of the new categories. More details on the decision tree algorithm itself will be presented in the future section of this chapter.

Continuous features value transformation can also be performed with help of decision trees similarly to what was discussed in the previous paragraph. The process of transforming continuous predictor into a discrete one is called ***discretization***. Apart from the decision tree based discretization, popular alternatives are ***equal frequency*** and ***equal width*** discretization. Equal frequency aims to create numbers of bins which contain the same number of customers. The frequency of a bin is calculated simply as:

$$bin\_size = \frac{n}{b} \qquad (1)$$

Where *b* is a number of bins desired by a modeler. After sorting a continuous variable x in ascending order and then give the same label to customer every *bin_size* customers.

Equal width discretization creates a number of bins *b* which contain even ranges of a continuous predictor. The width of such a bin is calculated as :

$$\Omega = (x_{max} - x_{\min} \quad)/b \qquad (2)$$

And similarly to equal frequency method, sorted list of customers by a variable x_i gets the same label every Ω of x_i[30].

2.2.2.2 Value Representation Techniques

After applying respective value transformation methods, depending on prediction algorithms to be employed, features need to be transformed into a suitable form. Coussement et al. (2017) distinguish three of them:

Dummy coding creates *v–1* features, while *v* being the number of distinct categories featured by a discrete covariate. Typically resulting dummy variables are binary, taking on only values

[30] Ibidem, p. 27-29.

of 1 and 0, where 1 indicates the presence of a category of a variable in a particular sample, while 0 its' absence. Ultimately this technique increases the number of variables in a training dataset D by *v–1* for each predictor on which this method is used. Although necessary in cases of certain algorithms (mainly logistic regression and any other regression-based frameworks), a drastic increase of D poses a lot of problems, such as increased learning time, harder interpretation, worse generalizability ultimately leading to employment of methods further reducing the number of original features or their distinct categories[31]. Next two methods try to minimize the negative impact of multiplying dummy variables by replacing categories of the dependent discrete feature with numerical values. These methods not only show their strength as an alternative to classical dummy coding but also greatly benefit any classification algorithm which prefers numerical input instead of a categorical one. This is due to the fact, that they base their values on the discriminative ability of dependent variable, something that Coussement et al. (2017) called in their overview of data preparation treatments (DPT) as ***churn link***[32].

First of them, ***incidence replacement*** simply translates each category of a feature into a proportion of churners. Following equation details how to calculate incidence replacement value for a predictor's group:

No of churners in a category / number of observations in a category (3)

The Weight of Evidence (WOE) is used as many business settings, therefore can and should be applied also in client churn prediction context. It can be seen as an extension of incidence replacement, however it replaces proportion with a log-odds equation of the following form:

ln(proportion of churners in a category / proportion of non-churners in a category) (4)

WOE takes on negative values if the proportion of non-churners is larger than churners and positive otherwise. The bigger the result of (4) the better the discriminative power of a category. Apart from benefits eliminating dummy coding drawbacks, WOE introduces non-linear logarithm transformation which can be preferable on its' own and also automatically makes the distribution of resulting random variable less skewed[33]. Another benefit of WOE is its' wide application in other business settings and the possibility to use it in a more advanced

[31] Ibidem, p. 30.
[32] Ibidem, p. 31.
[33] Ibidem p. 34-35.

way (such as regrouping categories with similar WOE value to further cut down on distinct categories number), which can be looked up by a practitioner in respective literature.

Although this step does not seem to bring much of a business value in terms of better churn reason understanding, possibly apart from methods of applying churn link, it can offer a dramatic improvement in predictive power. Although this step is often neglected not only by modeling teams but also by client churn prediction scholars, Coussmenet et al. show that its' studious application can dramatically improve prediction accuracy[34], allowing simple logistic regression model to even surpass advanced ensemble techniques. This line of research will be elaborated upon in chapter 3.

2.3 Modeling

After construction of a database, classification algorithm has to be selected. Of course, a practitioner does not have to limit herself to just one method, as multiple can be considered. However to select the best performing method a careful validation and input selection has to be performed. In this section popular classification algorithms, feature input selection, model training and evaluation methods are presented, with some of them coming directly from client churn prediction literature.

2.3.1 Selection of the Modeling Technique

As noted in the first chapter, classification algorithm has to be able to output a list of probabilities, along which the marketing department would be able to sort priorities of contacting customers with incentives. Other conditions it has to satisfy in order to maximize return on investment is its general availability and acknowledgment by the data mining community as a state-of-the-art algorithm. Therefore, the presented set of methods was established by arbitrary choice based on a review of client churn prediction literature under these constraints. This list is not exhaustive, but it is adequate for purposes of the experiment verifying study's research hypothesis.

2.3.1.1 Logistic Regression

Logistic regression is a classical statistical classifier modeling probability of a sample belonging to a category (e.g. customer or non-customer). It has essentially the same structure

[34] Ibidem p. 35-36.

as linear regression used to estimate a quantitative outcome by fitting a curve with the lowest mean squared error (MSE). However due to the fact that using classical regression for modeling this probability would result in non-sensible outcomes lower than 0 and greater than 1, fitting a straight curve simply against the probability of a sample belonging to a class is not an option. This can be illustrated by a simple example involving one independent variable. With *X* being the balance account, linear regression would try to find straight line minimizing MSE between *p(X)* and actual *Y* (taking only values of 0 and 1) with intercept *β* and slope β_1:

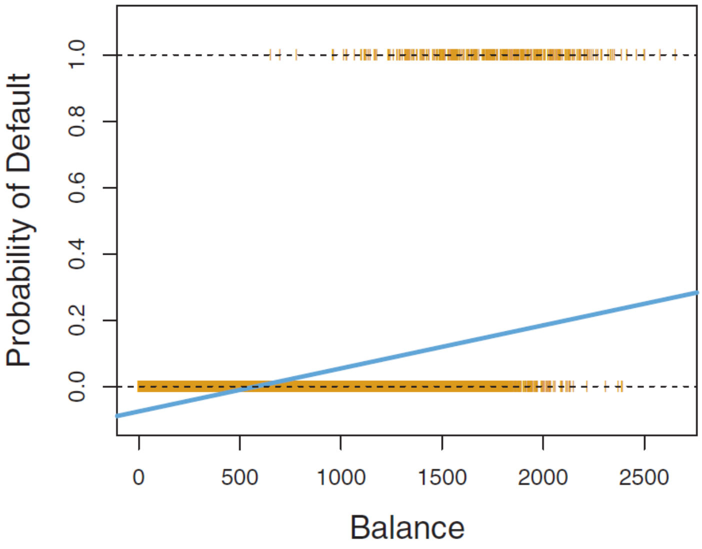


Figure 4. Fitting an ordinary linear regression to a classification problem. Source: James Gareth, Witten Daniela, Hastie Trevor, Tibshirani Robert (2013). An Introduction to ... op.cit., p. 131

As illustrated in figure 4, the predictive power of a regular linear regression would be underwhelming at best. In order to use the regression approach the curve should be estimated using a function that would always take on values in (0, 1). Due to its desired S-shape, *logistic function* is used instead, hence the name logistic regression. Then function estimating the *p(X)* curve takes on the following form:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} \qquad (5)$$

However as this form is unwieldy to use and interpret, after some algebraic transformations one can arrive at the following form:

$$log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X \qquad (6)$$

With left-hand side being often called logit or log-odds (as it is a logarithm of odds of belonging to a class). To estimate the intercept and coefficients, they have to be chosen in a way that maximizes the following ***likelihood function*** *l*:

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i`:y_{i`}=0}(1 - p(x_{i`})) \qquad (7)$$

Where x_i is a probability of positive sample belonging to a class estimated with equation 5, while x_i` being the probability of negative sample belonging to a class. A detailed description of optimization of the maximum likelihood function is out of the scope of this study, however very rarely a practitioner would need to understand it fully, as virtually all popular statistical software does it automatically. After plugging in both β_0 and $\beta_1$to the equation 5, we get an S-shaped *p(X)* estimate that is closer to 0 for values of *X* belonging mainly to non-defaulting customers ($y_i = 0$) and 1 to ranges of dependent variable belonging to mostly defaulting ($y_1 =$ 1) clients:

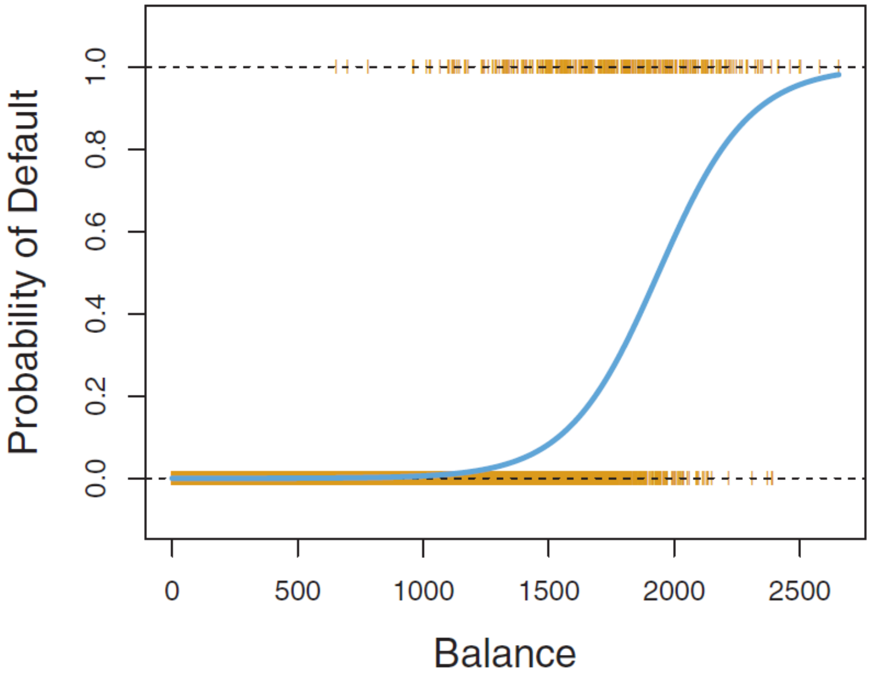


Figure 5. Fitting the logistic regression to the same classification problem as in figure 4. Source: James Gareth, Witten Daniela, Hastie Trevor, Tibshirani Robert (2013). An Introduction to … op.cit., p. 131

Apart from being well known and studied, the logistic regression method has another advantage that is highly desirable by client churn prediction professionals. Its log-odds form offers a similar interpretation to linear regression coefficients. Each increase in a unit of a predictor X_i increases the logit of belonging to a class by its estimated coefficient $\beta_{i.}$ This holds true only for β_i greater than 0, $p(Y = 1)$ drops with positive unit change if β_i is lower than 0, and does not influence this probability if the respective coefficient is 0 or close to it[35]. There are regularized versions of logistic regression aiming to suppress or eliminate features which estimated coefficient is not large enough[36]. Prediction is also straightforward – it is enough to just plug new samples dependent features values into the estimated equation. This gives a clear understanding of which and how individual factors influence attrition affinity, while also allows reducing the unnecessary features easing implementation of the model. For each coefficient, its standard error, confidence intervals, z-statistic used to test the null-hypothesis of it being a zero (meaning that a feature potentially has no influence on modeled probability), further strengthen interpretability of logistic regression[37]. All of the above is of course applicable also in case of having more than one independent variables. Although visualization of the estimated curve becomes impossible and procedure of maximizing likelihood function becomes more complex, other advantages, such as coefficients' meaning and usage of statistical tests to estimate confidence intervals are still available.

Although considered by churn prediction scholars as a valid benchmark algorithm[38] and widely used in practice[39], logistic regression makes a lot of assumptions which are often not feasible to fulfill with noisy marketing data. First of all, S-shape of the estimated curve works really well if there is a clear concentration of positive samples at one edge of the range of predictors, and negative at another. Strong bias of the model does not allow it to construct highly non-linear shapes to better generalize consolidations around certain covariate ranges. As already argued in the previous section, regression models have problems dealing with qualitative predictors with more than 2 levels. In this case, one of the levels has to be chosen as the baseline and only remaining levels are included as dummy variables in the model. Each dummy variable coefficient then represents the positive/negative impact on logit of a

[35] James Gareth, Witten Daniela, Hastie Trevor, Tibshirani Robert (2013). *An Introduction to Statistical Learning with Applications in R*. New York: Springer, p. 131- 137.
[36] Ibidem, p. 214-228.
[37] Ibidem, p. 148.
[38] Vafeiadis T., Diamantaras K.I., Sarigiannidis G., Chatzisavvas K.Ch (2015). A comparison … op.cit., p. 2-3
[39] Coussement Kristof, Van den Poel Dirk (2008). Churn prediction in… op.cit., p. 316.

variable's class against the arbitrarily chosen baseline. Baseline's impact is then indicated by the slope coefficient. With many discrete multi-level variables this becomes very confusing to interpret and data preparations treatment already discussed have to be employed. Furthermore including the interaction between variables is not straightforward as in other classification methods like decision trees. It also assumes no correlation between independent variables, so to ensure maximal predictive power, decorrelation of predictors becomes an important step[40].

2.3.1.2 Decision Trees

A ***decision tree*** is a class of classifiers aiming to successively divide predictors' space of all samples in a way that maximizes homogeneity of resulting smaller, and mutually exclusive groups with regard to the dependent variable. These divisions then create a set of simple decision rules one can follow to obtain a prediction for a sample[41]. These rules can be then visualized in a way which resembles an inverted tree, hence the name of the algorithm.

[40] James G., Witten D., Hastie T., Tibshirani R. (2013). *An Introduction to* … op. cit., p. 228-238.

[41] Vafeiadis T., Diamantaras K.I., Sarigiannidis G., Chatzisavvas K.Ch (2015). A comparison … op.cit., p. 4.

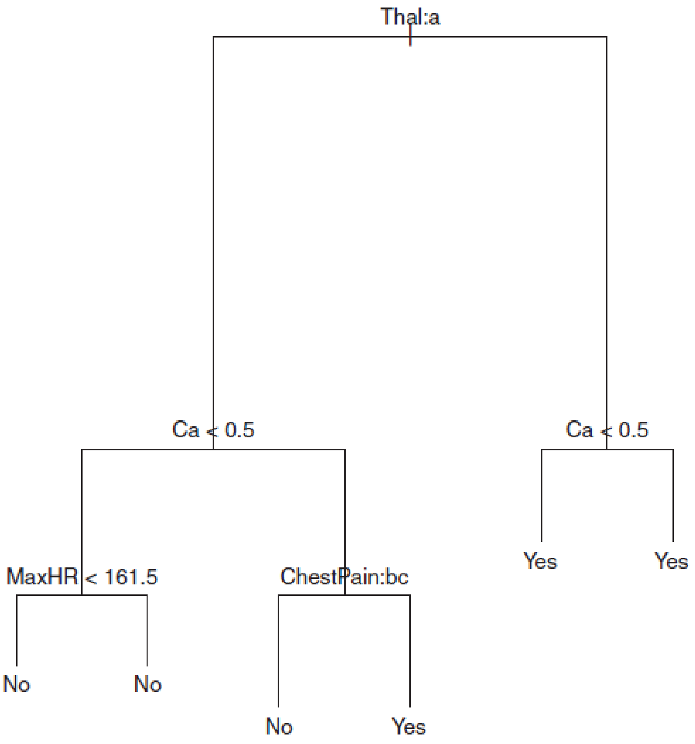


Figure 6. A decision tree visualization. Labels represent a dataset's features names and rules for a sample to belong to the left-hand node. Source: James G., Witten D., Hastie T., Tibshirani R. (2013). An Introduction to … op. cit., p. 313

However evaluating the homogeneity of all possible splits in features' space is not computationally feasible, especially on datasets with many continuous covariates. Therefore a simplifying, *greedy* approach is used, known as *recursive binary splitting*. It starts with identifying the best single split on one variable for all training samples. Then, for each resulting group, (*leaf* or *node* in decision tree parlance), the same process is applied. Procedure's greediness lies in the fact that it decides on a binary split only at the time of separation for each group, it does not consider any potential future splits. There can be different metrics employed to evaluate the goodness of a division. The simplest measure is just a fraction of observations not belonging to the most common class in a node, called the ***error rate***. The bigger the error rate, the less attractive a split is. As error rate lacks the increased level of sensitivity and makes fine-tuning divisions cumbersome, many decision tree implementations use the *Gini index*[42]*:*

$$G = \sum_{k=1}^{K} \hat{p_{mk}} (1 - \hat{p_{mk}}) \qquad (8)$$

k is an index of one of *K* classes, *m* represents one node of a tree. Not only it is much more granular than the error rate, but it also considers variation across all possible outcome classes. Small values indicate a better split, and often interpretation is that the smaller the Gini index, the ***purity*** (observation come principally from a single category) of a node is greater.

Numerous advantages made decision trees very popular not only in churn prediction literature but generally as a benchmark classification algorithm. Due to the nature of binary splitting, decision trees are robust against outliers, as they are simply considered as values between which a split may be performed. Dealing with a non-linear separation boundary between classes is also something decision trees handle quite well. As they divide the feature space into multidimensional rectangle shaped-regions, virtually any decision boundary can be drawn, similarly to how computer monitor uses square pixels to represent any shape given big enough resolution. Correlation between predictors is also a non-issue as each feature is considered individually for each separation. The scale of continuous and number of categories in qualitative covariates is also something that decision trees handle gracefully, without any need for special transformations. Probably the most important virtue in a business setting is their high interpretability. Decision tree graphs can be intuitively read by even non-numerical savvy employees, allowing even for prediction by hand. The graph can be enriched with node purity and count to enable drawing additional insights.

Unfortunately, numerous advantages are accompanied by a substantial amount of disadvantages. First of all decision trees are highly unstable due to the nature of recursive binary splitting. Even one observation can alter the resulting splits, in extreme cases changing variables and boundary values in each stump. Consequently, in contrast to the logistic regression which has a high bias, trees have high variance, hence are prone to overfit the training set. If not penalized, a tree will grow too deep to output very homogenous, but at the same very small groups, giving no generalization power. Therefore ***decision tree pruning***

[42] Caigny Arno De, Coussement Kristof, De Bock Koen W (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research*, 269, 2, p. 765

strategies have to be used to prevent these shortcomings. A popular one is to limit the minimum size of a resulting leaf, to 5% of training observations, where this number is a rule of thumb and based on specific need can be altered[43]. Pruning tools which are employed while growing a decision tree are often called ***stopping criterions***. There are also ***post-hoc*** methods, such as growing a tree as much as possible and then employ ***cost-complexity pruning***, which tries to find a smaller sub-tree with reduced complexity while preserving a satisfactory level of prediction accuracy. However number of possible subtrees to consider can grow substantially, rendering cost-complexity pruning complex, and stopping criterions do not guarantee effective defense against overfitting. Although decision trees have a lot of appeals, their use has to be careful if they are the only method employed.

2.3.1.3 Random Forest

Random forest is a modern state-of-the-art classification algorithm, which became very popular not only in client churn prediction settings but in many other, even non-economic branches of science such as bioinformatics[44]. Even though the random forest is based on decision trees, it introduces additional steps, mitigating their drawbacks. Firstly, to minimize the variance of growing a single tree, a ***majority vote*** between many fitted trees is used as the final prediction. Majority vote simply means, that predicted class for a sample is obtained by selecting the most popular class among results from each participating algorithms. To further reduce instability and increase generalizability, each tree in an ensemble (popularly referred to as *forest*) is fitted on a ***bootstrapped*** sample of the training dataset[45]. Bootstrap is a general resampling technique, used to estimate the uncertainty of virtually any estimation. Its idea is simple – as generating more samples from a true population a researcher wishes to base her statistic on is impossible (one cannot suddenly "find" all possible churning and non-churning customers for a company), we can "emulate" this population by randomly selecting existing samples with replacement[46]. At this point, the last crucial element is still missing, which adds "random" to the name of the technique. Even when growing many trees on bootstrapped samples, if there is a single predictor with high discriminative power on its own, there is very high chance that each tree's recursive binary splitting would begin with the same variable.

[43] Caigny A. D., Coussement K., De Bock K. W. (2018). *A new hybrid*... op. cit., p. 765.
[44] Larivière Bart, Van den Poel Dirk (2005). Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications*, 29, 2, p. 473.
[45] Ibidem, p. 474.
[46] James G., Witten D., Hastie T., Tibshirani R. (2013). An Introduction to … op. cit., p. 187.

This typically entails that the rest of each decision tree would look similar. If all of them are alike, using majority vote is not justifiable. Random forest introduces additional randomness, at each split in each tree only *D^(1/2)* variables are considered. This number can be altered, as long as a smaller subset then all *D* variables is used. However, *D^(1/2)* is an established default by method's author and is rarely deviated from[47]. This randomness effectively *decorrelates* trees in an ensemble, eliminating a scenario in which one strong covariate dominates split patterns. Apart from a number of predictors considered when creating trees (which essentially ale steers how big a single tree can become), a number of trees to grow is another parameter to choose by a professional. Typically a large number, between 500 and couple of thousands is selected, as the algorithm converges at the point when enough combinations of splits over bagged samples were generated to find all relevant patterns in a data set, while all mechanisms employed efficiently defended against overfitting.

Random forest advantages are numerous, but the main of them is excellent prediction accuracy, which lead to its popularity[48]. It also inherits many of the desirable characteristics of decision trees such as no need for transformation of variables despite their type, automatic handling of outliers, no need to eliminate correlated features and indifference to a different scale of predictors. Strong protection against overfitting and reasonable computing power requirements seem like a proverbial cherry on top of the cake when juxtaposed with an already impressive list of virtues easing industrialization.

It seems its weakest point, especially in a business setting, is decreased interpretability capability. Prediction for an unseen sample has to be done numerically in a statistical software on a trained model, as tree-like visualization is not feasible. However, even on this front random forest holds its ground as one of the most robust classification techniques. As a byproduct of performing so many splits on bagged samples, there is a possibility to calculate by how much a Gini index decreases on average across all trained trees when splitting node on a variable. In this way, one can calculate ***variable importance*** measure for each feature, which indicates which predictors have the highest discriminative power. There exists another way to measure the relative strength of predictors using ***out-of-bag*** (OOB) samples. Out-of-bag samples are the ones that did not make it into the bagged samples from the original training dataset. For each grown tree in the forest, these OOB observations can be used to

[47] Ibidem, p. 475.

[48] Ibidem, p. 476.

calculate prediction accuracy and store it. Then for a predictor of interest, its values in OOB vectors are randomly permutated, to break any potential correlation with the dependent variable. Prediction accuracy for trees involving selected predictor is calculated again, and if the decrease in accuracy is negligible, it means that it does not hold much of a predictive power. The average decrease of accuracy for a feature across all involved trees is sometimes referred to as *permutation importance* and is often reported together with the Gini index based by statistical software[49]. For both of them the higher the score, the more important a variable is. These metrics have to be always interpreted in relative, not absolute way. Their ranking is often similar, but permutation importance tends to distribute its score more uniformly. This way another virtue of random forests is revealed – with their easy implementation and robust importance measures, random forests can serve as a variable selection technique. However, these measures are not free from flaws. In the setting of many covariates, where some predictors might not be considered enough times due to randomization to fully measure their impact, when no clear splits exist on a variable for binary splitting and when there are heavy biases in bagged samples, are all examples of situations where trusting importance measures might not be reasonable[50].

[49] Hastie Trevor, Tibshirani Robert, Friedman Jerome (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition. New York: Springer, p. 593.

[50] Wright Marvin N., Ziegler Andreas, König Inke R. (2016). Do little interactions get lost in dark random forests? *BMC Bioinformatics*, 17, 1, p. 2.

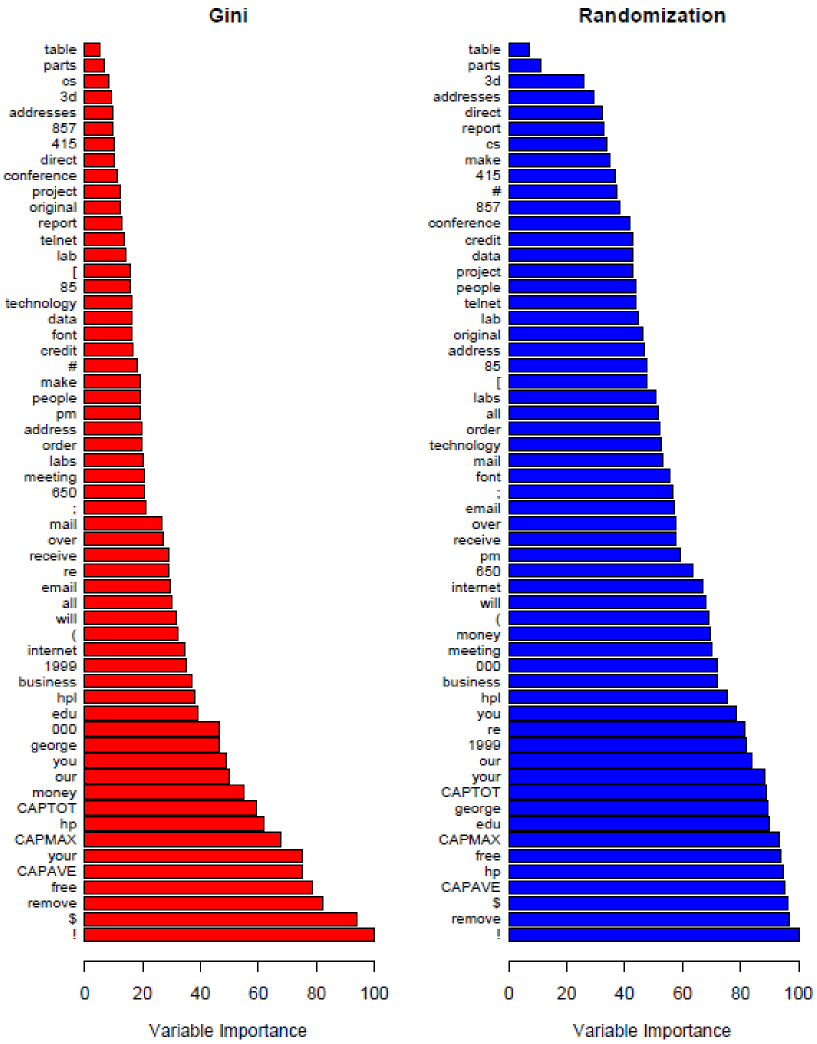


Figure 7. Random forest variable importance for problem of detecting spam emails. Source: Hastie Trevor, Tibshirani Robert, Friedman Jerome (2009). The Elements of … op.cit., p. 593

2.3.1.4 Support Vector Machines

Support Vector Machines is quite novel classification technique, which is very popular in binary classification problems, although it has extensions allowing for multi-class prediction. It is based on the initial idea of constructing *p-1* (where *p* is a number of dimensions in a dataset) dimensional ***separating hyperplane*** which separates observations according to their classes. Ideally, this plane should perfectly divide observations from both classes, while preferably being as far away as possible from points from opposing classes. ***Maximal margin hyperplane***, where the margin is the sum of perpendicular (i.e. minimal) distances from training points to the hyperplane, is the one which maximizes this gap. As can be seen in figure 8, this margin maximizing hyperplane shape and slope depend on the few closest points from each class. Unless the highlighted three points are moved (or farther one moves closer to the hyperplane than these three), maximal margin classifier will be insensitive to other observations movements[51]. Hence the dashed lines represent ***support vectors*** as they cross data points which decide the position of the maximal margin classifier and define the width of the margin.

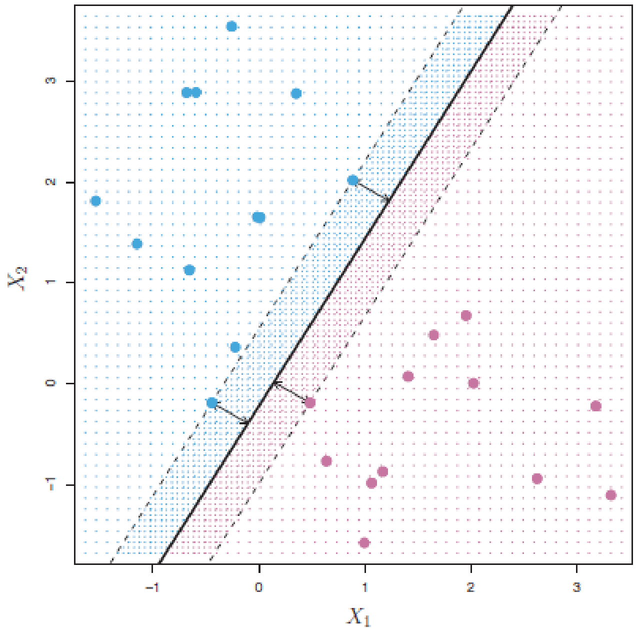


Figure 8. Support Vector Classifier example. Source: James G., Witten D., Hastie T., Tibshirani R. (2013). An Introduction to … op. cit., p. 342.

However, in the prediction setting, where the model is trained on and used on different data sets, this method does not guarantee generalizability and is prone to overfit a particular collection of observations. Furthermore with noisy data, and marketing data is considered as such, classes often cannot be separated perfectly with a hyperplane, even non-linear one. In such a scenario, a ***support vector classifier*** shall be used, which in essence is an extension of maximal margin classifier, however using ***soft*** margin instead of the maximal one. Soft margin allows for points of a class to fall on the incorrect side of a hyperplane. In greater detail, construction of such a classifier requires solving following optimization problem:

[51] James G., Witten D., Hastie T., Tibshirani R. (2013). An Introduction to … op. cit., p. 337–342.

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{maximize}} \quad M \tag{9}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \tag{10}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \tag{11}$$

$$\epsilon_i \geq 0, \ \sum_{i=1}^{n} \epsilon_i \leq C, \tag{12}$$

M signifies margin width, which in this problem should be maximized under the constraints listed in (9). The first constraint ensures that margin is based on observations with the shortest perpendicular distance from the separating hyperplane. The second equation is typically used to define a hyperplane, however then the right side of the inequality is simply equal to 0. This is because, contrary to most other classification techniques, negative observations (here non-churners), are represented as –1 instead of 0. Then equation (10) guarantees that each observation falls on the correct side of the hyperplane – y_i can be either –1 or 1, then the hyperplane equation has to result with a number falling on the same side of 0 as class label. *M* is used instead of zero on the right side if one wants to find coefficients of maximal margin classifier, with *M* preferable as big as possible. Soft margin classification adds ***slack variables*** ($\in_1$) together with nonnegative, tuning parameter *C* to the set of equations. To obtain a slack variable for an observation, one has to determine whether an observation falls on the correct side of the margin. If yes, then $\in_1 = 0$, if not then $\in_i > 0$, while values greater than 1 should only be attributed to the observations lying on the incorrect side of the hyperplane. *C*, however, limits how many of these mistakes support vector classifier can make. With *C = 0* the solution of this optimization problem results in maximal margin classifier. The bigger the *C*, the more such a model is volatile to new data points, hindering its generalizability by allowing increased variance[52]. Its' value has to be optimized numerically with cross-validation. While using soft margin, observations determining the position of the hyperplane are the ones that lie either on the margin or are on the incorrect side of it. Figure 9 shows

[52] James G., Witten D., Hastie T., Tibshirani R. (2013). An Introduction to … op. cit., p. 344-347.

results of decreasing *C* and visualizes how careful a researcher has to be when designing a tuning procedure to balance variance-bias trade-off.

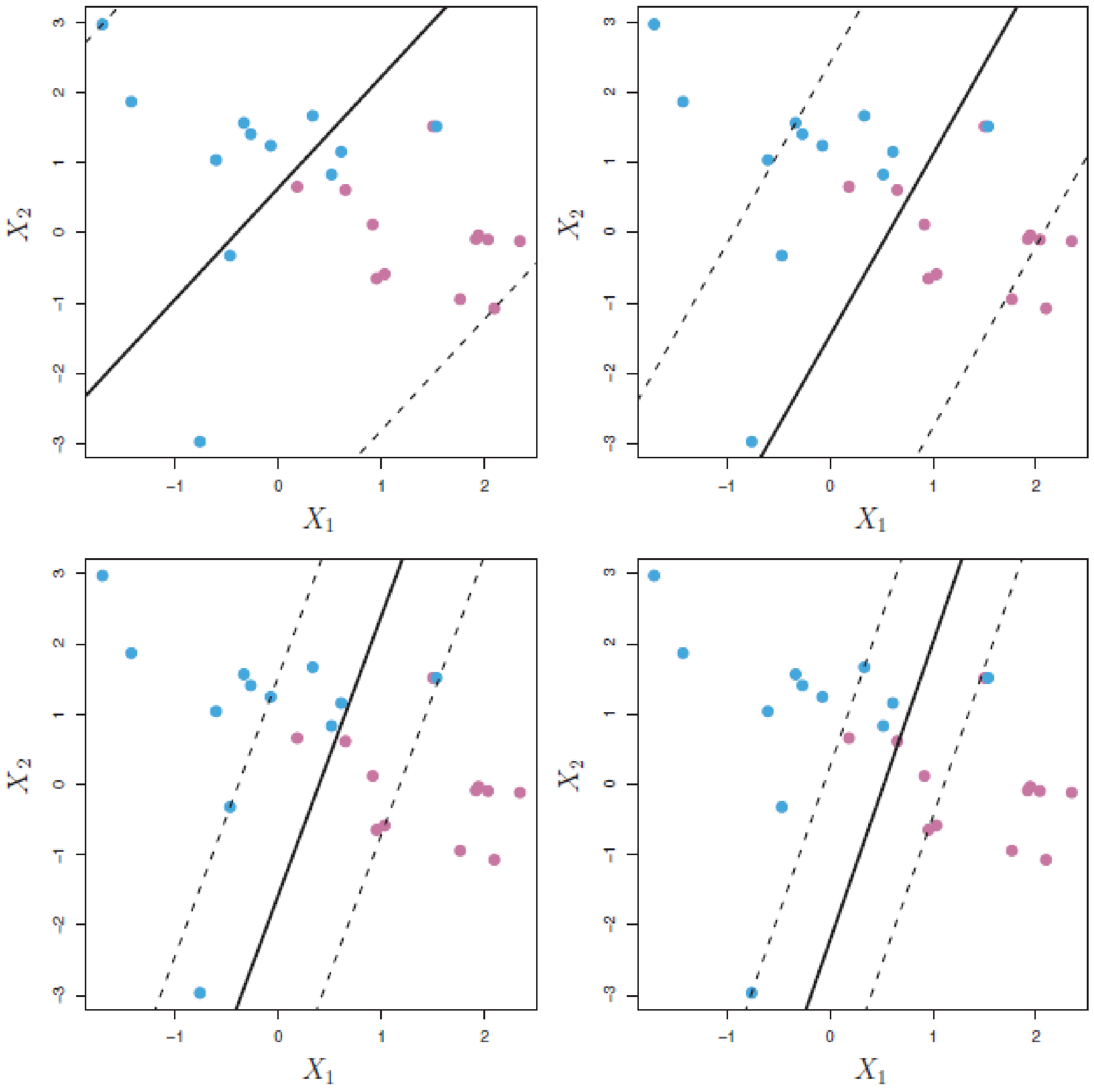


Figure 9. Support vector classifier fit on the same data using different values of C. The greatest value of C can be observed in the top-left corner, while the smallest in top-right. Source: James G., Witten D., Hastie T., Tibshirani R. (2013). An Introduction to … op. cit., p. 348.

Support Vector Classifier's most unique feature is probably its ignorance for observations lying far away from separating hyperplane when constructing it, which would be normally considered by other methods, affecting their discriminative capability. This is only true however when a linear decision boundary with a margin approximates the true boundary of the target population. ***Support Vector Machines*** (SVM) are then an extended version of support vector classifier, allowing for enlargement of dimensionality in hope of finding a separating hyperplane in the increased dimensionality space. A simple example of this would be adding polynomials of independent variables to the hyperplane equation and solving with it previously discussed optimization problem. However, as new features' structure and/or the number of added polynomials are limited solely by a practitioner's imagination, processing the problem with such an approach might lead to computations that are infeasible, even with today's computing power. SVM uses ***kernels*** to ensure that such an extension can be calculated efficiently. Although the description of solving support vector classifier optimization problem was not presented, it is possible to solve it just by using the inner products of all data points. This way, instead of maximizing margin over a set of n observations, one can perform it on *n(n-1)/2* pairs of training data points. In the setting of this study, an inner product for a pair of 2 observations is calculated as such[53]:

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j}. \qquad (13)$$

Using ***kernels*** allows for greater extensibility of this idea and gives a researcher greater freedom in shaping the non-linear decision boundary. The kernel is essentially a function that is evaluated on all distinct pairs of data points and replaces the inner product equation. The hyperplane equation (which is the target variable classification function) is then of the following form:

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i). \qquad (14)$$

The so-called ***kernel trick***[54] has yet another useful property allowing for further reduction in necessary calculations. Coefficient α_i becomes non-zero only for observations with indices being part of S, where S represents indexes of points falling on the support vectors. Simply

[53] James G., Witten D., Hastie T., Tibshirani R. (2013). An Introduction to … op. cit., p. 351.

[54] Coussement Kristof, Van den Poel Dirk (2008). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications*, 34, 1, p. 315.

using inner product formula as kernel, gives one a more efficient way to calculate support vector classifier. Two popular kernels for classification problem are a polynomial and radial kernel (producing a circular boundary in the original feature space):

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d. \tag{15}$$

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2). \tag{16}$$

Where d is a non-negative integer representing the desired order of the polynomials, and γ is a non-negative kernel parameter to be selected via cross-validation techniques.

SVM has several advantages which make them attractive in customer churn prediction setting. Firstly they guarantee excellent prediction performance, earning a well-deserved place among top off-the-shelf classification algorithms[55]. Possibility to choose among many kernel functions and the conscious trade-off between variance and bias when choosing the value of the parameter *C* gives a practitioner possibility to tune the method according to the problem at hand. However a radial kernel is advised by Coussement and Van den Poel in customer attrition setting due to its’ ability to handle non-linear relationship (in contrast to linear kernel) which are present in marketing customer data, while keeping computational cost at bay (in contrast to high degree polynomials, where values in enlarged space may go to infinity). Computing requirements are important, as one of the downsides of SVM is need to perform a costly grid search over an arbitrarily selected range of *C* and any additional kernel parameters. The biggest hit takes, however, interpretability, as SVM is a “black-box” algorithm, which does not offer practically useful interpretation capabilities[56].

[55] Ibidem, p. 316.

[56] De Bock Koen W., Van den Poel Dirk (2012). Reconciling performance and interpretability in customer churn prediction using ensemble learning based on generalized additive models. *Expert Systems with Applications*, 39, 8, p. 6819.

2.3.1.5 Logit Leaf Model

Logit Leaf Model (LLM) is a classification model designed specifically for client churn prediction[57] proposed recently by De Ciagny, Coussement, and De Bock (2018). It aims to increase the interpretability of a model while maintaining (at least) a similar level of prediction power to benchmark methods presented already in this chapter. In essence, it is an ensemble model composed of regression trees and logistic regression. With difficult marketing data, an analyst cannot expect all attributes to have linear or non-linear relationships with target variable, therefore a mixed approach seems like a good match. Logistic regression can handle linear dependencies well, while is not suited to model complex interactions between variables. This is something decision tree can accommodate, therefore the first phase of constructing an LLM model is to segment data according to terminal nodes of a decision tree. Then logistic regression is applied to each resulting, more homogeneous, group of customers to model linear relationships. Prediction is also straightforward. After fully training LLM model, one just needs to pass the sample through decision rules of the tree to attribute a new sample to the correct segment, then apply leaf's logistic regression coefficient to get the attrition probability. An interesting aspect of the method is that tuning parameters for J4.8/C4.5 decision tree algorithm used as part of LLM are selected via cross-validation when whole LLM model is constructed, not when constructing the inner tree. Authors offer implementation which uses both, pre-pruning in form of minimal leaf size as a percentage of training dataset and post-pruning in confidence threshold, which calculates a pessimistic, upper bound of the error rate of a node (the smaller the value, the heavier the pruning process). This way, fitting just a J4.8 decision tree would yield different values for these parameters after grid search, then when tuning whole LLM. This ensures that leaf segments are grown up to the point when they can be “handed off” to logistic regression for uncovering actionable insights in linear relationships with target variable. Tuning inner logistic regression models is performed with predictor ***forward selection***. Starting from a model containing only intercept, up to the model containing all possible predictors, at each step one variable is added. The one which addition results in the highest value of maximal likelihood function is retained before evaluating the model with an additional variable. This

[57] Caigny Arno De, Coussement Kristof, De Bock Koen W (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. European Journal of Operational Research, 269, 2, p. 760-772.

way one ends up with D + 1 models, and the best one is selected with ***Akaike Information Criterion***, defined as:

$$\mathrm{AIC} = 2k - 2\ln(\hat{L}) \qquad (17)$$

where *k* is a number of coefficients and *L* represents the value of the maximum likelihood function for a model. As smaller models are preferred, a model with the smallest AIC is chosen[58].

De Ciagny, Coussement, and De Bock (2018) designed a comprehensive experimental study on 14 churn datasets to showcase the effectiveness of LLM. When contrasted with random forests, singular C4.5 decision tree, logistic regression and more complex, but a related logistic model tree, its averaged rank overall 14 datasets was the lowest. This holds true for both model evaluation metric involved (top decile lift and AUC, which are discussed in next sections). Although it is not surprising to see an ensemble method outperform consistently its components, to gain the lead over much more robust and established random forest deserves attention. The biggest advantage of LLM, especially in a business setting, is comprehensibility. Output visualization of the model allows an analyst to quickly understand how customers can be segmented into more homogeneous groups, and which features drive churn for each of them[59]. This is not surprising as both components of LLM excel at interpretability. It is worth noting that the number of leaves grown within LLM is much lower than in a tree grown individually. Inheriting benefits of two methods, unfortunately, doesn't come free of negative heritage. Most notably all limitations applicable to pre-processing of features for logistic regression apply to LLM.

Seg	1st Step: Decision Tree		2nd step: Logistic Regression Coefficients					Segment Specific Coefficients	
	Rule 1	Rule 2	Intercept	retcalls	changem	credit	setprem	eqpdays	directas
1	eqpdays <= -0.31		0.46	1.12	-0.07	-0.28	-0.63	0.23	-0.52
2	eqpdays > -0.31	eqpdays <= -0.06	0.19	0.98	-0.24	-0.5	0.51		

Table 1. Example output visualization of the LLM model. Source: Caigny Arno De, Coussement Kristof, De Bock Koen W (2018). A new hybrid … op. cit., p.771

[58] Ibidem, p. 765.

[59] Ibidem, p. 772.

2.3.2 Input Selection

Even if a training dataset contains a set of business validated predictors, a modeling team should always strive to further limit the number of predictors used for selection and evaluation of the model. The motivation behind this logic is twofold. Firstly, the smaller the number of covariates, the easier it is for any algorithm to converge faster (or sometimes at all) and with less computational resources needed. Secondly, it is simply cheaper to gather, clean and update data for fewer variables on regular basis[60]. Also, savings on the operational level are introduced, because it is easier for analysts and practitioners to derive insights from model visualizations with fewer variables. However, input selection is not limited to selecting features, but also observations. This is of crucial importance in churn modeling where defecting customers constitute for a small fraction of a data set, rarely more than 20% and often up to a couple of percent of the number of data points. With such heavy skew, an analyst has to accommodate additional steps, most often in form of under and over-sampling, to overcome a model's tendency to over-fit in favor of the dominant class.

2.3.2.1 Variable Selection

Methods for variable selection are manifold. They can be mainly divided into two categories: ***filter methods*** which concentrate on one feature at a time, often taking into account its relationship with the dependent variable, and ***wrapper methods*** aiming to find the optimal combination of predictors to use in a model. Usually, it is more practical to apply a filtering approach first, as wrapper procedures complexity rise exponentially. It is out of the scope of this study to present a comprehensive set of them, however, methods featured by customer attrition prediction researchers will be highlighted[61].

Filter methods usually aim to calculate a score for each variable of a training set individually, create a sorted list and select a cut-off point, typically to include 20-50 variables for further processing and modeling. Filter methods are often based on correlations between the target and dependent predictors, the correlation between features to eliminate or unify related covariates, subset's intercorrelation or use regularized regression approaches suppressing/eliminating variables with the smallest statistical significance[62]. Nevertheless,

[60] Coussement Kristof, Lessmann Stefan, Verstraeten Geert (2017). A comparative analysis … op. cit., p. 30.
[61] Verbeke W., Dejaeger K., Martens D., Hur J., Baesens B. (2012). New insights into .. op. cit., p. 220.
[62] Ibidem, p. 220-223.

client churn prediction scholars have taken a liking to a very simple and popular yet effective method of preliminary variable selection – ***Fisher score***[63], popularly known as F-score*:*

$$Fisher\ score = \frac{|\bar{x}_C - \bar{x}_{NC}|}{\sqrt{s_C^2 + s_{NC}^2}} \qquad (18)$$

where *c* suffix signifies churning observations while *nc* loyal customers. Fisher score is higher for variables with better discriminating ability. An analyst can incorporate a more complex algorithm for selecting a cut-off point (such as cross-validating a preliminary model)[64], a predetermined number of variables or selecting a cut-off number by eye on elbow plot is often employed[65]. Main disadvantages of F-score are that categorical variables have to preprocessed as numerical ones (or as dummy variables)[66].

However filter methods have one big inherent disadvantage, namely, they do not take into account the interaction between variables. To remedy this, wrapper methods apply a heuristic to compare the performance of many combinations of variables in hope of maximizing the chance of finding the best performing one. One of them was already introduced when presenting mechanism for tuning logistic regression part of LLM *(forward stepwise selection).* ***Backward stepwise selection*** works similarly, but gradually excludes variables from a model involving all of them. The first step, after constructing a model involving all variables, is to evaluate the selected algorithm on all combinations of D-1 independent features. The variable which was not present in the best performing model at this stage is essentially excluded from further evaluation. This procedure is repeated until the one variable model is determined. The best performing modeling technique is then selected either via the maximum value of the evaluation metric or with some kind of score additionally penalizing number of features, such as already explained AIC.

Both of these frameworks aim to reduce the computational complexity of the ***best subset selection****,* which simply evaluates models involving all possible combinations of predictors. As Verbeket et al. showed in their study, best subset selection should not be aimed for in churn prediction, even in cases when there are no computational barriers. Despite datasets

[63] Caigny Arno De, Coussement Kristof, De Bock Koen W (2018). A new hybrid … op. cit. p. 765.
[64] Coussement Kristof, Van den Poel Dirk (2008). Churn prediction in … op. cit, p. 320.
[65] Coussement Kristof, Lessmann Stefan, Verstraeten Geert (2017). A comparative analysis … op. cit., p. 32.
[66] Coussement Kristof, Van den Poel Dirk (2008). Churn prediction in … op. cit, p. 320-322.

involving hundreds of variables, peak performance was achieved at around 8 variables. Fewer features simply were not providing enough information to generalize sufficiently, while exceeding 10 of them meant decreasing and ultimately negative marginal returns due to noise brought by additional predictors. Although this recommendation should be applied to other contexts than client attrition with caution, it clearly contradicts recent trends enticing businesses to gather as much data as possible, and authors highlight the importance of data quality over quantity[67].

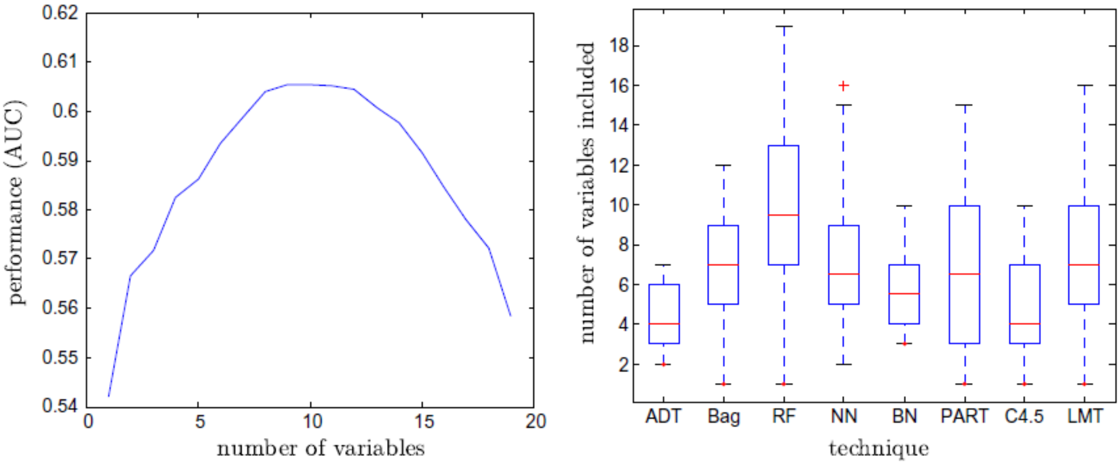


Figure 10. Left: model performance in function of a number of variables included in the model. Right: No matter what technique is used for churn prediction, the average number of variables used in the best model over 11 datasets is not bigger than 10. Source: Verbeke W., Dejaeger K., Martens D., Hur J., Baesens B. (2012). New insights into ... op. cit., p.223

2.3.2.2 Subsampling (Under/Over-Sampling)

As already explained, churn prediction datasets are heavily skewed towards non-churning observations, simply due to the fact that business creating economic value cannot have more attiring customers than regular paying ones. This poses a great challenge to all modeling techniques, as when only a small fraction[68] of values indicate one of the classes being predicted, the simplest and very effective generalization rule is just to predict all observations as renewing customers. Although there exist many strategies to remedy this problem, such as using modified versions of known classification techniques[69], applying weights penalizing

[67] Verbeke W., Dejaeger K., Martens D., Hur J., Baesens B. (2012). New insights into ... op. cit., p. 226.
[68] Ibidem p. 213.
[69] Xie Yaya, Li Xiu, Ngai E.W.T., Ying Weiyun (2009). Customer churn prediction using improved balanced random forests. *Expert Systems with Applications*, 36, 3, 1, p. 5449.

greatly misclassification of churners[70] or advanced observation „surrogate" sampling techniques such as SMOTE[71], customer churn researchers typically apply simple over or undersampling. ***Undersampling*** involves randomly removing observations from dominating class (churners) until the ratio of observations from opposing classes will achieve the desired level (commonly the upper bound and the most often choice is an equal split). ***Oversampling*** achieves the same outcome, however this time by copying churning observations as many times as necessary to achieve a balanced training dataset. Test set should always represent the original contribution. Both techniques do not add any new information, they just make relationships in less represented class more explicit.

Although over/undersampling has its allures as a technique which should solve the problem of class imbalance in an elegant and practically simple way, there is little evidence to support their usage by default. Verbeke et al. (2012) evaluated oversampled and regular versions of over 20 models on 13 datasets and they did not detect any statistically significant difference, reporting very varying effects from slight improvements to even degrading performance with oversampled versions[72]. Van den Poel and Coussment (2008) report no significant change in churn prediction using SVM on under-sampled training sets versus original class distribution, while even reporting increased performance with top decile lift evaluation metric on original skewed allocation[73]. These researchers continued this line of study, this time on a greater number of algorithms. Although under-sampling brought much better results for the random forest in half of the reported datasets, for other algorithms there was no significant difference, advanced CUBE sampling method did not perform better and there was no clear recommendation on which artificial split of class distributions helps to remedy the class imbalance problem. Despite this, authors argue that undersampling should have an edge over oversampling. Increasing the count of minority class introduces new information, however, generalizability could suffer as an algorithm might start creating rules involving overfitting same samples occurring multiple times. Analysts should decide on a case by case basis, giving slight preference to simple undersampling[74]. There also exists another practical way to achieve more balanced samples. Churner definition can be somewhat relaxed, as depending

[70] Burez J., Van den Poel D. Handling class imbalance in customer churn prediction (2009). *Expert Systems with Applications,* 36, 3, 1, p. 4630.
[71] Ibidem, 4632.
[72] Verbeke W., Dejaeger K., Martens D., Hur J., Baesens B. (2012). New insights into ... op. cit., p. 226.
[73] Coussement Kristof, Van den Poel Dirk (2008). Churn prediction in …, op. cit. p. 318.
[74] Burez J., Van den Poel D. Handling class… op. cit., p. 4632.

on the cost structure of an enterprise a customer who stays with the company for some time, or brings profits over a certain threshold, even if ultimately ceased to be a client, sustained company's profitability in a substantial way[75].

2.4 Training and Evaluation of the Model

In order to guarantee a good generalizability and avoid over-fitting model to a dataset, firstly it is divided, usually randomly, into train and testing subsets. Numbers of samples in each being, typically, respectively 2/3 and 1/3 of the original database. All steps discussed so far, such as data preprocessing, selection of variables and performing a stepwise selection of predictor combinations are applied and evaluated on the train set. The final evaluation of the model is performed on the test set, which has to be composed of samples never seen during the training of the model. This ensures that the real-life scenario is emulated when the model has to cope only with yet unseen samples. To impose even more rigorous experimental scenario, an additional validation set can be introduced, based on which model tuning decisions are evaluated before continuing with the final assessment on the test dataset. When tweaking a model and assessing its final performance on the test set, an ***evaluation metric*** is needed for establishing the best model. This section describes ***cross-validation*** procedure which aims to find the best algorithm parameters during its training phase and then introduces the evaluation metric used to assess the model's efficacy under customer defection prediction paradigm.

2.4.1 Cross-validation

Cross-validation is a procedure that is used in order to select optimal parameters for a model while ensuring that the chosen set of values will guarantee similar performance on the testing set. More generally, ***k-fold cross validation*** consists of several steps. Firstly, a training dataset is divided into k mutually exclusive folds. Then a model is trained on k-1 folds with a chosen set of parameters, while the kth fold is held out for testing and calculating value of evaluation metric chosen[76]. In order to complete procedure for a particular model, each of k folds has to be used once as training set resulting in k evaluation metric values. In order to assess a model's strength mean of these k values is recorded as a performance measurement for a particular set of parameters. This average aims to ensure that the influence of atypical

[75] Ibidem, p. 320.

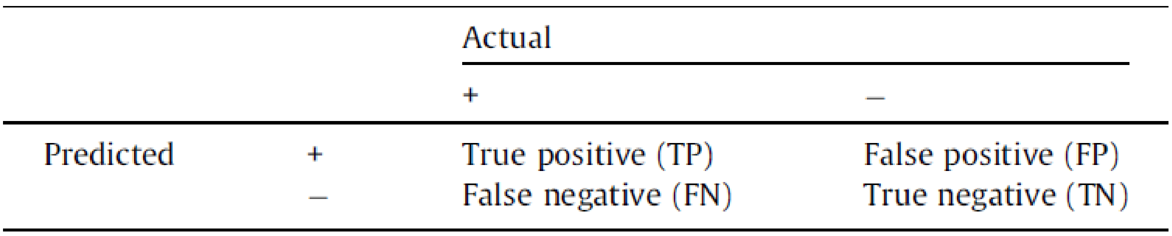
[76] Larivière Bart, Van den Poel Dirk (2005). Predicting customer … op. cit., p. 20.

observations is reduced, decreasing the variance of the model's strength estimation and that this score on unseen samples will be similar. As most of the discussed classification algorithms have free parameters which are by default set with recommended, but ultimately arbitrarily chosen values (such as number of predictors considered at each split in random forest or misclassification budget C in SVM), an analyst should use ***grid search*** of many (or all if feasible) possible combination of these parameters to find the most optimal one. Each round of grid search consists of training a model with a particular set of parameters using cross-validation. A popular choice in the data mining community is to perform 10-fold cross-validation. Moreover, a single cross-validation procedure can be repeated many times in order to further reduce variance and improve the reliability of the averaged score. Churn prediction scholars use 5 times repeated 2-fold cross validation (5 x 2 CV) as it has better statistical properties (with several test statistics developed for it) and is better suited for unbalanced samples[77].

2.4.2 Evaluation Metrics

2.4.2.1 Confusion Matrix Based Metrics

One can represent the output of prediction of a binary classification algorithm on a dataset in the following table, called ***confusion matrix***:



		Actual	
		+	–
Predicted	+	True positive (TP)	False positive (FP)
	–	False negative (FN)	True negative (TN)

Table 2. Confusion matrix for binary classification problem. Source: Burez J., Van den Poel D. Handling class …, op. cit., p. 4627

where ***true positives*** (TP) is a number of correctly identified positive samples (churners), ***true negatives*** (TN) is a number of correctly indicated negative observations (non-churners), while ***false negatives*** (FN) and ***false positives*** (FP) represent the amount of incorrectly predicted truly positive and respectively negative samples. These 4 values allow one to calculate many popular evaluation metrics for classification methods:

[77] Burez J., Van den Poel D. Handling class …, op. cit., p. 4633.

$$\text{specificity} = \frac{\text{TN}}{N} \Rightarrow 1 - \text{specificity} = \frac{\text{FP}}{N} = \text{FPrate}$$

$$\text{sensitivity} = \frac{\text{TP}}{P} = \text{TPrate} = \text{recall}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{P + N}$$

$$\Rightarrow \text{misclassification error} \quad (\text{MER}) = 1 - \text{accuracy}$$

of which accuracy (also known as percent correctly classified) is the most widely used. Although present by default in virtually any statistical software, these measures have 2 main disadvantages which make them obsolete in churn modeling. Firstly these metrics attribute the same *misclassification cost* to false negatives and false positives. Misclassified true churners, (false negatives) are way more costly to the company than clients who were predicted as churners but turned out to extend their commitment. This allows reporting often 90% or greater accuracy, making a score of other measures equally attractive, which under normal circumstances would be an indication of an excellent classification algorithm. The second disadvantage is invariance to the chosen value of the cut-off point. As most algorithms output a probability of a sample belonging to the positive class, an analyst can choose virtually any value between 0 and 1 as the threshold under which samples are classified as non-churners. By default, it is 0.5, but depending on the scenario at hand, it might be more profitable to lower it. As these measures consider one threshold at a time, it is difficult to get a sense of how well an algorithm performs over a range of possible cut-off values[78]. As establishing misclassification cost and/or threshold value upfront is practically very difficult[79] to churn modeling researchers turn to other metrics to assess the power of a classification.

2.4.2.2 Area Under Receiver Operating Characteristic Curve

AUC, which is an abbreviation for Area under the Receiver Operating Characteristic Curve (ROC) battles both deficiencies of traditional evaluation metrics of binary classification. ROC is constructed by considering all possible cut-off threshold values for the output of a binary

[78] Ibidem, p. 4630.

[79] Verbeke W., Dejaeger K., Martens D., Hur J., Baesens B. (2012). New insights into ... op. cit., p. 230.

classification algorithm *(t)* and marking on x-axis *1-specificity(t) (*false positive rate), while on y-axis *sensitivity(t)* (true positive rate)[80].

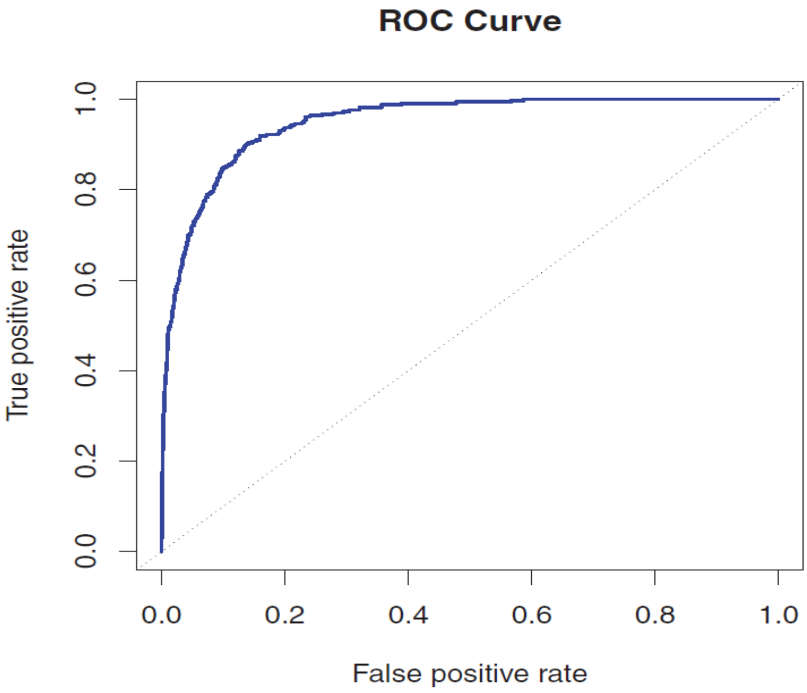


Figure 11. ROC Curve. Source: Burez J., Van den Poel D. Handling class …, op. cit., p. 4629

AUC takes on values between 0 and 1, while a random algorithm (guessing the outcome class) takes on the value of 0.5, represented on as diagonal on figure 11**.** Therefore the better performing the algorithm, the closer the value to 1. Intuitively AUC represents an averaged predictive power of an algorithm and expresses the probability that a random positive observation (churner) has higher or equal chances of being classified as a positive one then a random negative (non-churner) sample according to the studied algorithm. Thanks to this both issues of misclassification cost and threshold selection are avoided. Formally AUC is calculated as follow:

$$\mathrm{AUC} = \int_0^1 \frac{\mathrm{TP}}{P} \, \mathrm{d}\frac{\mathrm{FP}}{N} \qquad (19)$$

[80] De Bock Koen W., Van den Poel Dirk (2012). Reconciling performance and interpretability in customer churn prediction using ensemble learning based on generalized additive models. *Expert Systems with Applications*, 39, 8, p. 6820.

2.4.2.3 Top Decile Lift

Although AUC has good properties and statistical legacy (such as a test for significance of the difference between two ROC curves), its general purpose hinders its usefulness in churn modeling. Generally, CRM managers are not interested in using an algorithm that guarantees the best overall performance. As explained in chapter 1, incentive campaigns are usually employed on the small subset of the customer base due to costs, therefore marketing decision makers want to make sure that targeted part can be characterized by a big density of potential churners. ***Top decile lift*** is an evaluation metric designed to address this need, while at the same time avoiding deficiencies of classical confusion matrix measures. To calculate it, an analyst needs to sort posterior probabilities of the output of an algorithm in descending order (from most to least probably attiring observations) and then divide share of true churners in the top decile of that list by the total share of churning clients in the dataset **(π)[81]:**

$$Top - decile\, lift = \frac{\pi_{10\%}}{\pi} \qquad (20)$$

2.4.2.3 Maximum Profit Criterion

Although top-decile lift has a lot of traction among marketing managers, Verbeke et al. criticize it for two reasons. A top decile is an arbitrarily chosen range to compare the density of churners against and maximizing defectors density based on posterior prosperity to attire does not guarantee the most profitable retention campaign. This might lead to sub-optimal model selection, not from point of view of the modeling team, but from the perspective of overall CRM campaign success. As the ultimate goal of a for-profit organization is to profitably multiply invested capital, these researchers decided to link evaluation of churn model with profit maximization of a retention campaign. ***Maximum profit criterion*** (MP) is a unique churn model evaluation metric, estimating returns from a campaign based on it, while at the same time outputting the optimal fraction of customers to target[82]. Formally, profit from a retention campaign can be expressed as[83]:

[81] De Bock Koen W., Van den Poel Dirk (2012). Reconciling performance… op. cit., p. 6821.
[82] Verbeke W., Dejaeger K., Martens D., Hur J., Baesens B. (2012). New insights into ... op. cit., p. 214.
[83] Verbraken Thomas, Verbeke Wouter, Baesens Bart (2013). A Novel Profit Maximizing Metric for Measuring Classification Performance of Customer Churn Prediction Models. *IEEE Transactions on Knowledge and Data Engineering*, 25, 5, p. 965.

$$\Pi = N\alpha[\beta\gamma(CLV - c - \delta) + \beta(1-\gamma)(-c) + (1-\beta)(-c-\delta)] - A \quad (21)$$

where:

- N – the number of customers in the dataset
- α – a fraction of customers targeted in the retention campaign and offered a retention incentive
- β – a fraction of true would-be churners of clients included in retention campaign (determined based on classification's confusion matrix)
- δ – the cost of the incentive when the customer accepts it and stays
- γ – a success rate of the incentive, i.e. fraction of targeted customers who were would-be churners, but stayed due to incentive
- c – the cost of contacting a customer to offer him an incentive
- CLV – average customer lifetime value of a retained customer, i.e. average amount of revenue a customer will generate during their tenure
- A – fixed administrative cost of running a retention campaign

What is interesting about this formula, is that it takes into account not only losses incurred by a customer churning but also by customers who get essentially free incentive benefits while they were never inclined to attire. In this way, MP is able to maximize the density of true churners in the targeted fraction, while also keep the size of α in check, as in reality, a small but well-targeted campaign would yield biggest returns. Definition of MP criterion is as follow:

$$\mathrm{MP} = \max_{\alpha}(\Pi). \quad (22)$$

Another unique feature and unfortunately hindrance is the fact that certain parameters need to be known in advance. Notably γ, CLV, δ, and c. The success rate of an incentive campaign can usually be derived from historical data, while other parameters should already be something that a company stores. As total administrative cost, A is always the same constant amount, of no importance to the analysis, it can be discarded from classifier evaluation[84]. This is a novel and specialized model selection approach, as it changes the axis of analysis from numerical optimization to driving benefits for the business. Figure 12 visualizes benefits of selecting the model and optimal fraction to target with MP calculated by Verbeke et al. (2012) on 11 real-world datasets**.** The top-most gray area represents benefits over selecting a model with top decile lift. As two top horizontal lines, determining boundaries of this rectangle, were evaluated with methods providing a fraction of customers to target, they are parallel to the x-axis. Even if an analyst was somehow able to determine the optimal fraction of clients to target, selecting a model using AUC would result in almost 2.5 times lower profit per customer.

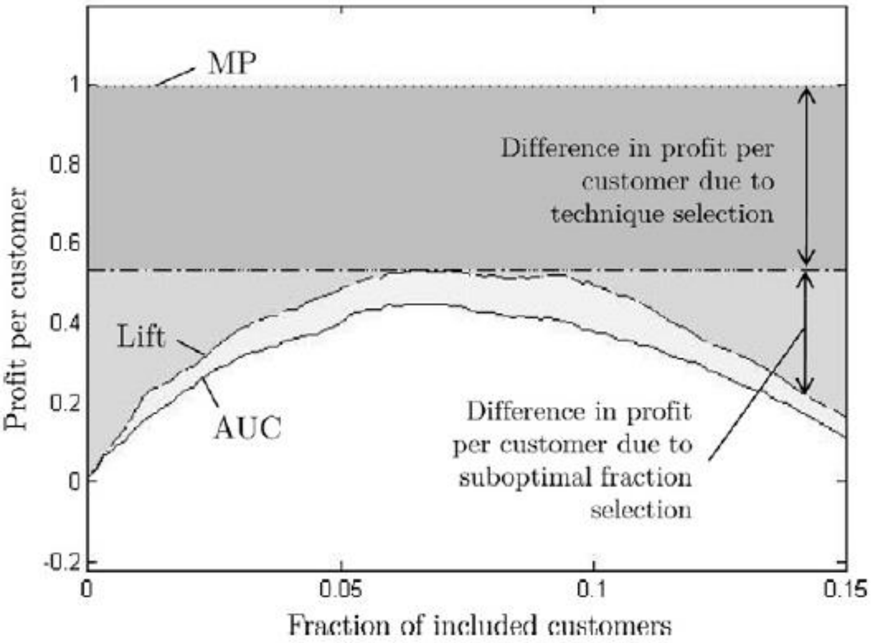


Figure 12. Benefits of using MP criterion as an evaluation metric. Source: Verbeke W., Dejaeger K., Martens D., Hur J., Baesens B. (2012). New insights into ... op. cit., p.226

[84] Ibidem, p. 967.

Chapter 3. B2B SaaS Experimental Case Study

This chapter presents the application of methods and theories from chapter 2 on a real-life dataset provided by a Software as a Service company serving other businesses and not individual clients (SaaS B2B). Unique circumstances of this kind of business together with study's contribution to churn prediction literature are discussed, afterward study's hypotheses are presented. Then, discussion of the experiment's findings which address these hypotheses is preceded by a description of experiment's design. This study is concluded with final remarks regarding study's findings, limitations and problems to explore by future research.

3.1 Research Setting and Hypothesis

The main contribution of this study is the application of recent methods and findings of client churn prediction literature described in chapter 2 in a new context of a SaaS B2B company. Software as a service B2B companies usually offer a computer software hosted on their own servers (popularly known as *cloud-based technology*), which was prohibitively hard to install on-premise and expensive to purchase, in a simple web browser interface for which clients are charged periodically a small subscription amount. This makes possible for small companies to deploy state of the art CRM, ERM or virtually any other kind of enterprise-class IT software, without the need to pay upfront for expensive contracts. Business to business setting already poses a difficulty to churn prediction methods as a number of clients is much smaller (typically counted not more than in thousands) as opposed to banking and telco providers with millions of individual customers. Additionally, the competitive SaaS landscape is fragmented, due to the proliferation of many strictly product-oriented start-up companies. Even though good customer attrition prediction model should contain variables related not only to the service usage but also marketing, financial and demographical information, these small players do not have respective departments developed enough to maintain huge data warehouses containing hundreds of diverse customer attributes. SaaS contract lengths are often much shorter, even monthly, while contracts signed with financial institutions and telco operators are of several years' length. Due to this B2B paradigm lessens one of the difficulties - class imbalance problem is not that severe as fluctuations in customer base are much more frequent.

With discussed context in mind, the following research hypothesis are addressed in the experimental case study:

1. Simple, more interpretable models, notably logistic regression and decision tree, trained on a dataset processed with data preparation treatments recommended by Coussement et al. (2008), are able to outperform across all used evaluation metrics more complex models (RF, SVM, LLM) trained on dataset cleansed in a typical way.
2. Algorithm selected by Verbeke's et al. (2012) maximum profit criterion is different from the best one indicated AUC and top decile lift, solidifying it as the evaluation metric to employ in client churn prediction.
3. Logistic Leaf Model, as a model specifically crafted for client attrition forecast, performs at least as good as other algorithms across all evaluation metrics, and due to its enhanced interpretability should be the model selected for production use.

Study's hypothesis and experiment were designed in a way to give SaaS B2B practitioners a clear guidance on which modern churn prediction methods they should consider in their companies and whether using more specialized approaches, such as MP or LLM has potential to pay off. This set of hypothesis exposes yet another unique feature of this study, as many methods recommended by separate churn prediction research groups (LLM, MP, 5x2 CV, DPT) are combined together to draw recommendations. Other studies offer a comparison of one of these elements in contrast to generally known methods, thus not exploring interactions between them. It also aims to participate in ongoing debate whether using more interpretable models on carefully prepared data in a business setting can offer acceptable levels or predictive performance as complex ensemble methods while ensuring greater comprehensibility among co-workers. To strengthen the practical implication of this study, the experiment was conducted entirely using R statistical programming language[85], which is very popular free and open source software for data analysis and statistical modeling. All models, apart from LLM, were trained using R's caret package[86]. MP criterion[87] and LLM[88] were implemented using author's respective packages for these methods. Algorithms selected for the experiment are a subset of recommended churn prediction methods, which can be used

[85] R Development Core Team (2008). *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing.

[86] Kuhn, M. (2008). Caret package. Journal of Statistical Software, 28, 5, p. 1-26.

[87] https://cran.r-project.org/web/packages/EMP/index.html

[88] https://cran.r-project.org/web/packages/LLM/index.html

in practice and be understood without the need to look up much additional information from specialized literature, such as recently popular, yet complex in their architecture neural networks.

3.2 Experiment Design

3.2.1 Dataset

Data used in this study was provided by a company creating outbound sales automatization SaaS B2B tool. This service allows clients to efficiently browse a large database of potential customers and then launch a fully automated email campaign on selected contacts. It is a difficult dataset, with very dirty data and variables not falling into any specific statistical distribution or fall into power-law distributions. Due to confidentiality and data acquisition cost issues, this dataset has features mainly coming from a database of product's usage statistics. It is also a relatively small dataset, consisting of 2954 observations and 48 features. 34 of variables describe application usage statistics, 8 financial matters (such as contract amount), and 6 marketing features (such as client source). All features requiring lagging (e.g. no. of emails sent) are captured at around 80% of contract's length. For example, this means that for a yearly contract, variables are captured after 10 months from its start. In order to increase the size of the dataset and capture propensity to attire at different customer's lifetime stages, one observation in this dataset consists of a customer at around 80% of one particular contract length. 27.4 % of observations are churning data points.

3.2.2 Data Preparation Treatment (DPT)

Firstly, the dataset is divided randomly into training and test subsets, where the first one consists of roughly 80% of observations. Each set is balanced by the dependent variable, maintaining around 28% of churning observations in each of them. Although 80/20 split seems like a controversial one with such a small database, it was influenced by business setting, as then test set size is roughly equivalent to what CRM team can handle within one retention campaign, simulating very well models' real-life performance. Validation set approach was not used due to a modest count of data points. All DPT methods are first used on the training dataset and then employed on the test set using parameters established with the help of the training set. Handling of missing values was not necessary, as this dataset contains no unexpectedly missing values.

In order to assess hypothesis 1., three variations of training and test datasets are created. The first type, called *normal_dpt* follows an often preprocessing scheme of normalizing all continuous variables and dummy coding discrete predictors[89]. Only 6 features required dummy coding, with all of them having no more than four possible categories, avoiding sharp increase of covariate numbers. Continuous predictors in the test set were normalized using each feature's mean and standard deviation from the training dataset. Due to the fact that substantial amount of customer can be considered as outliers (either using service a lot or almost at all in comparative view), a business-driven decision was made to not employ additional outlier handling techniques for regular data preparation process. *special_dpt* is the second class in which optimal data preparation treatment reported by Coussement et al. (2008)[90] were used. These optimal methods were decision tree based remapping for categorical variables and equal frequency binning as value transformation steps, followed by Weight of Evidence conversion as value representation step. DT based remapping was not necessary, as categorical variables have a maximum granularity of 4 levels validated by business logic. Following Coussment's et al. (2008) recommendation 10 bins were used for equal frequency binning, then boundaries of these bins were used on the test set to determine each observation bin participation. In certain cases, it was not possible to determine boundaries of 10 divisions using the training set, due to not enough number of distinct values, so the biggest possible bin number was used. WOE conversion was then calculated for each bin on the test set to replace their labels. Bin labels in the test set were simply replaced with the corresponding WOE value from the training set. Final, third variation of both datasets, called *special_dpt_fisher* is simply a subset of variables of the special_dpt database. On the special_dpt training dataset, Fisher score was used to select 10 independent variables with the highest ability to discriminate between churning and extending observations.

3.2.3 Training and Evaluation of the Models

Five types of models (LR, DT, RF, radial kernel SVM and LLM) were trained on all 3 variants of training datasets resulting in a total of 15 fits. For models requiring selection of parameters (all without LR), 2-fold cross validation repeated 5 times was used. Each fold is also balanced based on the distribution of a dependent variable, containing around 30% of churning observations. Due to the skew of the predicted variable is not strong, with limited

[89] Coussement Kristof, Lessmann Stefan, Verstraeten Geert (2017). A comparative analysis ... op.cit., p. 30.
[90] Ibidem, p. 31.

evidence justifying the use of subsampling techniques, no other mechanism to defend against class imbalance was used. A full grid search is used to determine the best set of model parameters using cross-validation. Possible values for each setting were based on previous studies and are presented in table 3. Evaluation metric employed at this stage is AUC. After determining the best performing parameters via cross-validation, each model is trained one last time on the full training set with optimal settings from the previous step. Finally, each model's performance is evaluated on the test set using AUC, top decile lift and maximum profit criterion. Unfortunately, CLV cost of retention offer and the cost of contacting a customer, used in MP cannot be disclosed.

Classifier	Meta-Parameter	Candidate Settings
Decision Tree	Confidence Threshold for Pruning	.01, .15, …, .30
	Min. leaf size	n*[.01, .025, .05, .1, .25, .5]
Random Forest	No. of Trees	1100
	No. of randomly sampled variables	(D*[.1, .25, .5, 1, 2, 4]) ^ 1/2
Radial basis kernel SVM	Regularization penalty (C)	2^(-13, -12, …, 12)
	Width of kernel	2^(-13, -12,…, -1)
LLM	Confidence Threshold for Pruning	.01, .15, …, .30
	Min. leaf size	n*[.01, .025, .05, .1, .25, .5]

Table 3. Ranges of model parameters used in the cross-validation part of the experiment. Source: *Coussement Kristof, Lessmann Stefan, Verstraeten Geert (2017). A comparative analysis ... op.cit.* and own data.

3.3 Experiment Results and Findings

The table below present all algorithms test set performance across evaluation metrics:

Model	AUC			Top Decile Lift (TD)			Maximum Profit Criterion			MP Optimal Fraction to Target		
	normal dpt	special dpt	special dpt fisher	normal dpt	special dpt	special dpt fisher	normal dpt	special dpt	special dpt fisher	normal dpt	special dpt	special dpt fisher
LR	0.73	0.78	0.76	2.28	2.47	2.28	222.10	261.61	246.66	48%	45%	52%
DT	0.75	0.74	0.75	2.04	2.41	2.35	243.46	219.97	229.58	42%	55%	35%
RF	0.77	0.77	0.77	2.28	2.35	2.53	250.93	247.73	243.46	44%	47%	51%
SVM	0.74	0.79	0.76	2.35	2.35	2.28	217.83	269.08	233.85	54%	47%	45%
LLM	0.73	0.78	0.76	2.28	2.41	2.22	217.83	255.20	242.39	52%	51%	48%

Table 4. Results of the experiment for all algorithms and evaluation metrics involved. Source: own data.

Evaluation metric values indicate that usage of recommended data preparation treatments can boost quite substantially prediction performance of an algorithm. Logistic regression trained on a special_dpt dataset performs better, or equally as good, as any other algorithm trained on a normalized dataset across all metrics. Using MP criterion, the performance boost offered amounts up to 17%, while AUC improved by 7% when compared with SVM and LLM on normalized set. At the same time special_dpt LR is the most business-wise efficient metric offering almost the highest possible maximal profit while targeting the smallest fraction of customers in a retention campaign. The single decision tree does not seem to benefit from DPT when looking at AUC and MP, achieving substantial gains only when looking at TDL. With the relatively small number of observations, reducing the resolution of variables seems to work against a single DT. One can also see that RF is a powerful improvement over a singular decision tree, as changes in DP techniques do not influence much its performance across all measures, offering a consistent level of efficiency. Results indicate that DPT recommended for churn prediction problems by Coussement et al. (2008) should be employed, even on such difficult and small datasets as in this study. It is also evident that simple Fisher based feature selection approach is not enough to achieve optimal performance. Due to the scarcity of observations, ten variables is not enough to properly generalize. However, models with only 10 variables based on DPT treated datasets often offer an improvement in many cases, indicating Fisher score as a viable and simple way for prelaminar feature selection. It seems that a wrapper method employed on top of Fisher score can offer even greater amelioration. Noteworthy is fact, that employment of special_dpt increased average Fisher score of all variables from normal_dpt's 0.014 to 0.047.

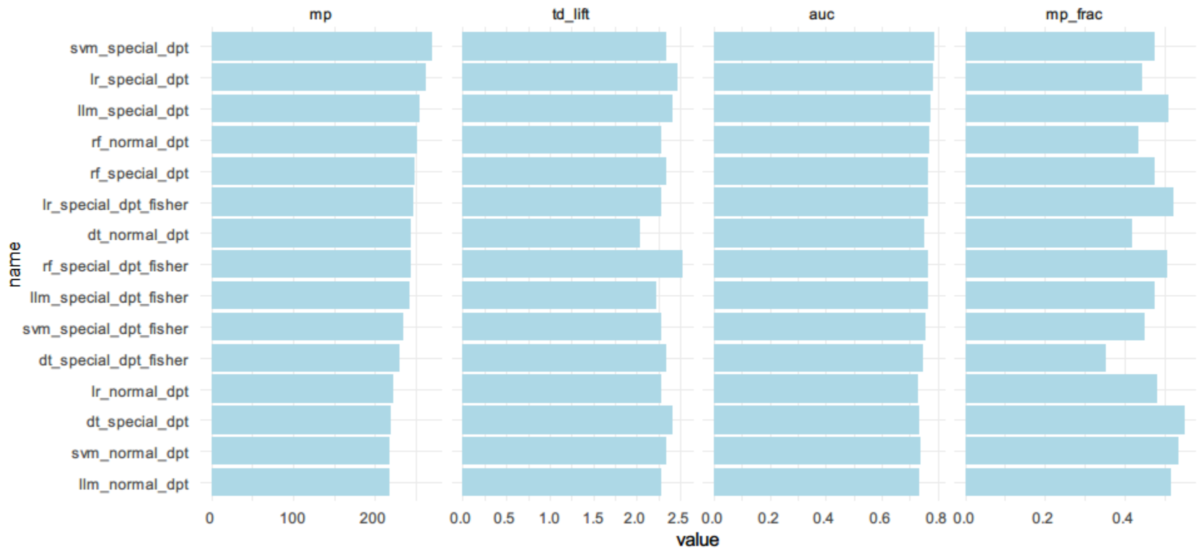


Figure 13. Table lens chart of test performance of all algorithms trained over all types of DPT. Source: own data.

Results of the experiment indicate that there is a strong positive correlation between AUC and MP. This is not a surprise, as algorithms performing well statistically also will have a tendency to yield more profit overall. Top decile lift, however, ranks algorithms quite differently from other two evaluation measures. Optimal MP fraction also indicates that first decile, which ultimately is arbitrarily chosen a value, is far from representing the optimal fraction of customers to include in a retention campaign. Taking into account fact, that test set in this experiment represents approximately size of a possible single retention campaign, selecting an algorithm with top decile lift would be misleading, resulting in a method which does not guarantee a maximum profit of such a campaign. What is more interesting, the optimal fraction of MP indicates that not even half of customers participating in a campaign have to be contacted in order to maximize a campaign's performance.

Algorithm	Average Rank
DT	3.44
LLM	3.22
LR	3.33
RF	2.11
SVM	2.67

Table 5. Average rank of each prediction algorithm over all types of data preparation treatment and all evaluation metrics. Source: own data.

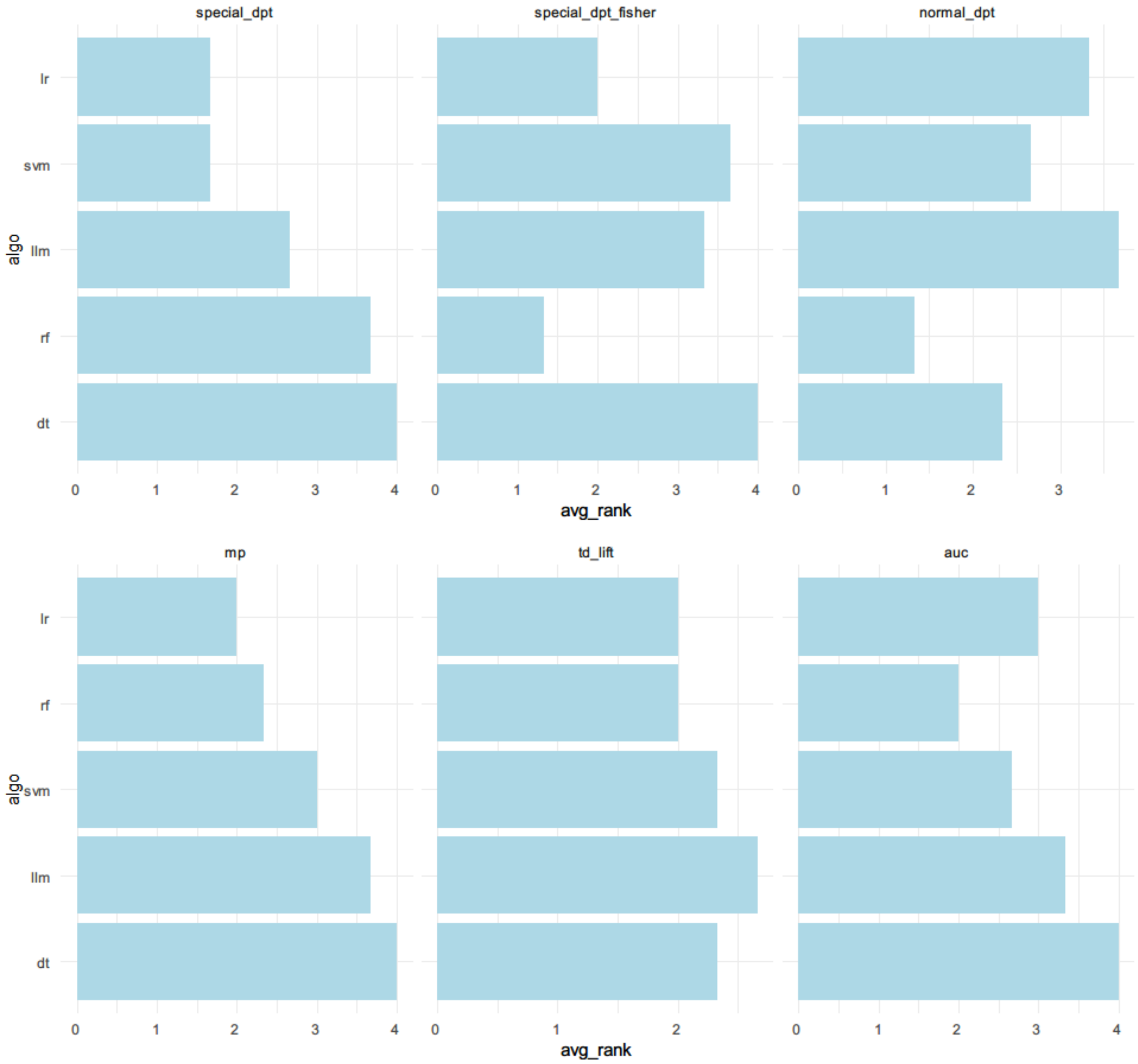


Figure 14. Table lens chart depicting overage rank of prediction algorithms over data preparation treatment types and evaluation metrics.

Finally, plots of rankings give insights into how algorithms perform in comparison to each other. To no surprise, more complex random forest technique performs best, while logistic regression is not far in terms of global average rank. LR good performance is attributed to big gains on DPT prepared datasets. Rankings across evaluation metrics are similar for single tree-based methods (DT, LLM) consistently occupying the last two spots. Ranks of algorithms across DPT method very much stronger, with already noted gains for special_dpt datasets, drastic drop in RF rank for full special_dpt database and decisions tree performing only better on the normalized dataset. It seems that Coussement et al. (2008) DPT techniques favor methods relying on individual features relationship with a dependent variable instead of interaction between variables. This was not unexpected, as WOE conversion employs churn-link of a single variable, disregarding interactions. RF is emerging as an overall leader, followed by SVM. RF has an advantage of being conceptually simpler than SVM and natural ability to tackle outliers, therefore serving as a great starting point to assess potential profitability and magnitude of efficacy of a modeling initiative in early project stages. LLM model globally outperforms only DT, but we can see it benefits greatly from the usage of DPT. Rankings, however, cannot depict the whole picture, as in terms of nominal MP value LLM model is almost the best one, overtaking RF. Although this study does not provide much evidence to support a claim of LLM's predictive power superiority, it cannot be discarded from modeling team's arsenal as strictly worse. Its ability to distinguish customer segments and churn drivers inside each is something that no other method can offer, making it a recommended way to model customer attrition classification problem.

Summary and Conclusions

The main goal of this study was to present to the reader the importance of customer churn prediction in current economic setting, its methods and processes in managing profitable long-term relationships with the customers and how modern data mining approaches should be used in an organization to successfully launch a retention campaign. In order to do so, the first chapter of this work discussed definitions and phases of CRM, together with types of data modeling which can improve the results of each of them. It concluded with indicating constraints that CRM context imposes on predictive modeling, such as the need for specialized model evaluation metrics, handling of class imbalance, the classification output which can be sorted by the propensity to attire and the increased interpretability of a model to aid in managerial decision-making.

The second chapter presented the theoretical underpinnings of data mining methods which are recommended for customer churn prediction. Firstly, the process of implementing data mining solutions in a business setting, CRISP-DM, was discussed. Then each phases' apparatus is presented. In order to offer the reader the most up to date recommendations, methods used by churn prediction scholars in the last dozen years are described. In the first stage, a database on which churn prediction model will learn upon should be composed in 40% of attributes describing a client's product usage, together with financial, marketing and socio-demographic features. After such a dataset was constructed, it has to undergo a data preparation treatment in order to maximize prediction performance, as well as transform data into a suitable input format. Categorical variables can be transformed with decision-tree base remapping, continuous ones with decision-tree based discretization, equal frequency or equal width discretization. Transformed variables can be then expressed as dummy variables, replaced with their incidence replacement value or converted to WOE. Next, four benchmark algorithms, namely logistic regression, decision tree, random forest, and SVM are depicted together with their shortcomings and advantages in client churn prediction setting. Additionally, a specialized LLM model, which comes directly from client attrition literature is discussed in the same way. Variable selection, then model evaluation and selection techniques are discussed next. The second chapter concludes with a description of appropriate model evaluation metrics. AUC and top decile lift are juxtaposed against maximum profit criterion, a unique measure which indicates the best performing models based directly on potential monetary return from a retention campaign.

The third chapter presents the study's research hypothesis. Experiment's goal was to verify whether DPT and variable selection methods can increase the efficacy of a classifier, test MP criterion as a viable evaluation metrics and probe on a preferred algorithm to use in production use. Methods described in chapter 2 are employed on a real-life dataset donated by a SaaS B2B company. Experiment's findings show that indeed DPT can increase the efficacy of a model, however, the boost was most significant for methods searching for linear relationships between predictors and target variable (logistic regression and LLM). Logistic regression trained on a special_dpt dataset performs better, or equally as good, as any other algorithm trained on a normalized dataset across all metrics. Looking at the MP criterion, the performance increase amounts up to 17%, while AUC improved by 7% when compared with SVM and LLM on the normalized set. MP turned out as an indispensable tool for churn prediction practitioners. Its results were close to AUC, meaning that it has the ability to discriminate between algorithms which perform better or worse statistically while offering explicit monetary profit for a retention campaign based on a model, as well as the optimal fraction of customers to target. As it was close to 50%, arbitrary 10% of top decile lift prevented it from indicating the most profitable method overall. Random forest resulted as the most versatile and prone to all kinds of data pre-processing, however, the absolute differences between maximum profits' of top performing algorithms and DPT treatments were not substantial. Due to limited interpretability of random forests, LLM emerged as a very attractive method for production use.

Although this study covered many topics, and combined recommendations from multiple churn prediction literature studies, it is not free of limitations. Due to the proliferation of SaaS startup companies, one dataset is not enough to draw fully generalizable conclusions and draw boundaries of this generalization. As class imbalance problems tend to not be so heavy for these companies, further research in validating present churn prediction methods for SaaS B2B companies would be most welcome. Another issue to tackle would be a modest size of datasets, which contrasts with millions of observations coming from telecommunications and financial industries on which presented methods were tuned. Even though this study's findings that as long as the employed algorithm is an established one, the prediction power remains on a relatively similar level between them, is in line with Verbeke et al. (2012)

research, some interesting interactions with Coussement et al. (2017) churn DPTs seem to have emerged. As they have a tendency to ameliorate performance mostly of methods not relying on the interaction between variables, such as logistic regression, exploring the interplay between different types of churn prediction algorithms and data preparation seems like a viable line of research.

Bibliography

Ballings Michel, Van den Poel Dirk (2013). Kernel Factory: An ensemble of kernel machines. *Expert Systems with Applications*, 40, 8, p. 2904-2913

Burez J., Van den Poel D. Handling class imbalance in customer churn prediction (2009). *Expert Systems with Applications*, 36, 3, 1, p. 4626-4636

Caigny Arno De, Coussement Kristof, De Bock Koen W (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research*, 269, 2, p. 760-772

Chen Zhen-Yu, Fan Zhi-Ping, Sun Minghe (2012). A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *European Journal of Operational Research*, 223, 2, p. 461-472

Coussement Kristof, Van den Poel Dirk (2008). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications*, 34, 1, p. 313-327

Coussement Kristof, Lessmann Stefan, Verstraeten Geert (2017). A comparative analysis of data preparation algorithms for customer churn prediction: A case study in the telecommunication industry. *Decision Support Systems*, 95, p. 27-36

De Bock Koen W., Coussement Kristof (2010). Ensemble classification based on generalized additive models. *Computational Statistics & Data Analysis*, 54, 6, p. 1535-1546

De Bock Koen W., Van den Poel Dirk (2012). Reconciling performance and interpretability in customer churn prediction using ensemble learning based on generalized additive models. *Expert Systems with Applications*, 39, 8, p. 6816-6826

Hastie Trevor, Tibshirani Robert, Friedman Jerome (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Second Edition. New York: Springer.

James Gareth, Witten Daniela, Hastie Trevor, Tibshirani Robert (2013). *An Introduction to Statistical Learning with Applications in R*. New York: Springer. ISBN: 978-1-4614-7138-7 (eBook)

Kim, C.W., Mauborgne, R. (2005). Blue Ocean Strategy: From Theory to Practice. *California Management Review*, 47, 3, p. 105-121

Kimball Ralph, Ross Margy (2016). *The Kimball Group Reader: Relentlessly Practical Tools for Data Warehousing and Business Intelligence*. Indianapolis: Wiley.

Kuhn, M. (2008). Caret package. *Journal of Statistical Software*, 28, 5, p. 1-26

Larivière Bart, Van den Poel Dirk (2005). Predicting customer retention and profitability by using random forests and regression forests techniques. *Expert Systems with Applications*, 29, 2, p. 472-484

Ngai E.W.T., Xiu Li, Chau D.C.K (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications* 36, 2, 2, p. 2592-2602

Provost Foster, Fawcett Tom (2013). *Data Science for Business.* Sebastopol, CA: O'Reilly Media. ISBN: 978-1-449-36132-7

R Development Core Team (2008). *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing. ISBN: 3-900051-07-0. URL: http://www.R-project.org

Tufte Edward R (2001). *The Visual Display of Quantitative Information*. Connecticut: Graphics Press.

Vafeiadis T., Diamantaras K.I., Sarigiannidis G., Chatzisavvas K.Ch (2015). A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*, 55, p. 1-9

Verbeke Wouter, Dejaeger Karel, Martens David, Hur Joon, Baesens Bart (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research*, 218, 1, p. 211-229

Verbraken Thomas, Verbeke Wouter, Baesens Bart (2013). A Novel Profit Maximizing Metric for Measuring Classification Performance of Customer Churn Prediction Models. *IEEE Transactions on Knowledge and Data Engineering*, 25, 5, p. 961-973

Wickham Hadley (2014). Tidy Data. *Journal of Statistical Software*, 59, 10, p. 1-23

Wright Marvin N., Ziegler Andreas, König Inke R. (2016). Do little interactions get lost in dark random forests? *BMC Bioinformatics*, 17, 1, p. 1-10

Xie Yaya, Li Xiu, Ngai E.W.T., Ying Weiyun (2009). Customer churn prediction using improved balanced random forests. *Expert Systems with Applications*, 36, 3, 1, p. 5445-5449

List of Tables

List of Figures