FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Kostenbasierte statistische Methoden zur Betrugserkennung

Vorhersage schlechter Bonität unter Beachtung des individuellen Risikos

MASTERARBEIT

zur Erlangung des akademischen Grades

Master of Science

im Rahmen des Studiums

Business Informatics

eingereicht von

BSC. Georg Heiler
Matrikelnummer 1225063

an der Fakultät für Informatik
der Technischen Universität Wien
Betreuung: Univ.-Prof. Dipl.-Ing. Dr. techn. Peter Filzmoser

Wien, 1. November 2017

Georg Heiler	Peter Filzmoser

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Cost-based statistical methods for fraud detection

Prediction of never paying customers considering individual risk

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Business Informatics

by

BSC. Georg Heiler
Registration Number 1225063

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.-Prof. Dipl.-Ing. Dr. techn. Peter Filzmoser

Vienna, 1st November, 2017

Georg Heiler	Peter Filzmoser

Erklärung zur Verfassung der Arbeit

BSC. Georg Heiler
Tokiostraße 17/308, 1220 Vienna, Austria

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. November 2017

Georg Heiler

Acknowledgements

This thesis concludes my master's degree in Business Informatics at the Vienna University of Technology. It was written at the Institute of Computational Statistics under the supervision of Univ.Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser with data and funding from our partner company T-Mobile Austria.

I would like to thank Professor Peter Filzmoser for his excellent guidance and assistance.

Thanks a lot, Georg Petzl for being my supervisor at T-Mobile for opening up doors to access data hidden in silos which otherwise would not have been possible for me.

Dear Mohamed Ibrahim, thanks for introducing me to essential details of T-Mobile's credit check process.

Dear Christoph Körner thank you so much for getting me in contact with such an exciting topic for a thesis and reviewing my code.

Dear Dieter Knittel thanks a lot for organizing all the little but so important things that I could concentrate on my thesis.

Dear Robert Fidler, thanks for helping me accessing and sanity checking the cost metrics.

Dear Markus Wiesinger, thank you for helping me to keep a clear vision for where the thesis should be heading and helping me build a network at T-Mobile Austria to understand better the business processes which generate the data.

Moreover, thanks to all the other people who helped me to understand the business processes and gather the data at sufficient quality.

Kurzfassung

Telekommunikationsunternehmen treten zunehmend als Finanziers der immer teureren Endgeräte auf. Mit einher geht die Gefahr von Zahlungsausfällen betrügerischer Kunden, die versuchen, Mobiltelefone, aber auch Dienstleistungen des Providers gratis zu nutzen, d.h. ohne jemals eine einzige Rechnung zu begleichen. Klassische Kreditschutzverfahren stellen keinen ausreichenden Schutz dar. Moderne maschinelle Lernverfahren können aufgrund einer besseren Segmentierung der Kunden wesentliche Vorteile bieten. Der Einsatz von speziell kostenbasierten Algorithmen können die Einsparungen noch erhöhen.

Wir vergleichen die Ergebnisse des klassischen Bonitätsprüfungsprozesses unseres Partners mit normalen und speziell kostenbasierten Verfahren des maschinellen Lernens. Hierzu entwickeln wir eine Kostenmatrix um das Risiko im Einzelfall optimal beurteilen zu können.

Abstract

Telecommunication providers not only offer services but increasingly finance consumer devices. Credit scoring and the detection of fraud for new account applications gained importance as standard credit approval processes showed to fall short for new customers as there is only scarce information available in internal systems. Modern machine learning algorithms, however, can still infer intricate patterns from the data and thus can efficiently classify customers. Cost-sensitive methodologies can even enhance the savings.

In this thesis, we develop a cost matrix which allows evaluating the individual risk of accepting a new customer and therefore helps to prevent new account subscription fraud optimally.

Executive summary

Many devices are lost as the standard credit check process is focusing on detecting defaults but falls short at detecting fraudulent or customers who never pay a single bill as only scarce information is present.

Machine learning can offer great possibilities to smarten business processes. Introducing the notion of cost and savings to the machine learning model can help to evaluate better the individual risk of accepting a single customer. We found that:

- machine learned fraud predictors can offer huge savings compared to the classical credit scoring process
- the strategy should be set out clearly in the beginning what the machine learning algorithm has to achieve and optimize
- before starting a data science project a data engineering project is required to build an appropriate data pipeline which offers timely & quality controlled access to the data
- make sure to dedicate IT resources to integrate the data science findings into the business processes

Contents

CHAPTER 1

Introduction

> A fraudulent phone call is one in which there is no intent to pay – theft of service.
>
> *Richard Becker, (Becker et al., 2010)*

The most significant losses in the telecommunication industry are caused by fraud, which means not paying for the usage of services (Wieland, 2004). Losses are estimated to amount to several billions of dollars of uncollectible debt per day (Moudani et al., 2013). In the industry, much data is regularly collected for billing purposes. However, in the past, it was not possible to analyze this amount of data efficiently to aid fraud detection. Now, a customer's fraudulent behavior can be analyzed on a bigger scale. Through the use of smart machine learning algorithms on ever-increasing quantities of data the exploration of more complex, but insightful patterns for the detection of fraud is possible. Most research for telecommunication data was focused on trying to identify if a customer turns fraudulent or insolvent over time (Daskalaki et al., 2003). Inherently, this has been a reactive approach.

Nowadays it is common for telecommunication providers to not only sell communication services but also support the financing of mobile phones. In case a new and unknown customer applies for a contract there exists no prior record in the companies IT systems regarding the customer's creditworthiness. In particular, this is a problem for Austria as there are lots of foreign people from the eastern part of Europe or, e.g., from Germany. Some cases are known where German citizens with a bad credit history from German credit scoring agencies can obtain expensive mobile devices as this data is not shared across country boundaries. Proactive steps are needed to prevent fraud efficiently as the most significant initial cost of subscription fraud is the cost for expensive mobile devices

accompanying the contract (T-Mobile Austria, 2016). This problem is particularly pressing for private customers; therefore, we will only focus on this type of customer group. A *neverpayer* is defined as a new private customer who did not pay a single bill within the first three months of a contract. Two types of neverpayers need to be differentiated:

- regular customers with financial problems
- criminals regularly stealing devices

As such one should note, that not every neverpayer is a criminal.

Usually, regular credit scoring agencies were relied upon to check if the customer is creditworthy. Such a scoring agency is used for our partner's credit check process, however, requires much manual effort to check all the warnings issued. Aggressive marketing of high-priced devices recently increased the ratio of losses.

As has been shown by (Wilcox et al., 2013) that a better score of creditworthiness can be obtained if more complex patterns in the data or new types of data like social networks or geolocations are considered, our partner decided to evaluate machine learning solution to help solve this problem. In particular, we were comparing the standard credit check process with regular machine learning models, state of the art classifiers and particular cost-based models to identify a good solution. For the cost-based models, we propose a unique cost matrix for the telecommunication industry.

As telephone services are considered a basic need in infrastructure, similar to water or gas, the provider is obliged to conclude contracts with subscribers due to this definition as a Universal Service as definden in §27 of the Federal Austrian Telecommunicaiton Act 2003 (TKG). As a consequence, it is not possible to terminate a telecommunication contract immediately, as defined in §70 leg cit. *The provider of Telecommunication is obliged to remind the subscriber of the missing payment, an early warning that the provider might interrupt or disconnect the service and as a last step to grant at least a periode of additional two weeks before the disconnection (this is also necessary to fulfill the legal obligation to offer the number portability also if the subscriber never paid a bill).* Therefore, as depicted in Figure 1.1, a considerable time lag is incurred until a fraudulent account can be closed. Such a regular dunning process takes up to several months. In case of a bad customer, initially, the price of the often costly hardware is lost. Due to expensive roaming interconnection fees between the providers during the dunning process much money can be lost in the consecutive months so that in total the sum of losses per fraudulent customer can increase by much.

There is a clear need in proactively scoring creditworthiness of new customers for postpaid services and loans or at least to recognize the fraudulent behavior as fast as possible to keep the timeframe of fraudulent usage, and thus the incurred loss low. Our partner's goal is to solve both problems by creating tailored algorithms for each case. Some deal with T_0, which indicates the starting time point, directly at the Point of sale (POS) to predict fraud and prevent losing the costly hardware, others only target T_x trying

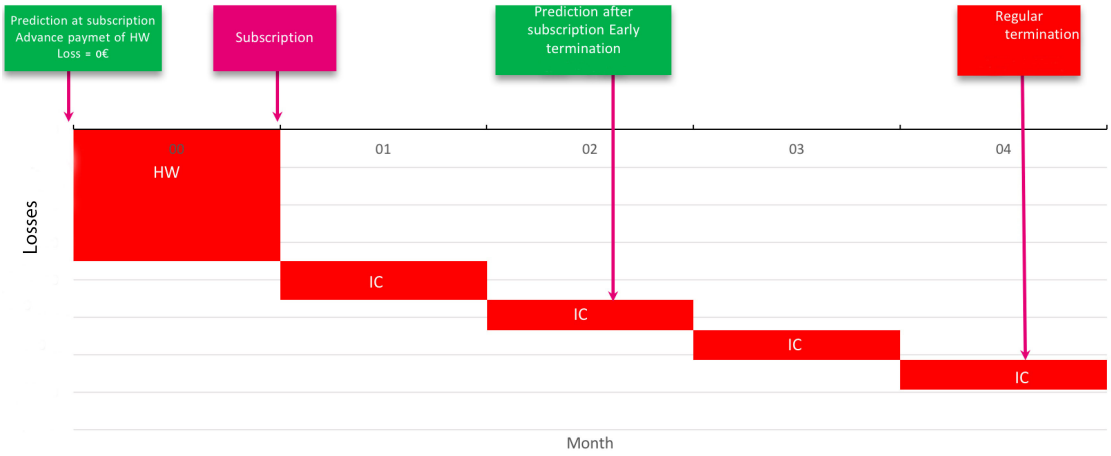


Figure 1.1: Proposal of loss reduction through early prediction of non paying customers and speedier dunning process (T-Mobile Austria, 2016)

to achieve early termination to prevent outstanding unpaid bills by roaming fees if the model did not catch the fraudster initially. However, this thesis will only analyze the approaches for T_0 as:

- data is readily available
- the largest savings are possible if loss of hardware is prevented in the first place
- it is easier to integrate into business processes
- the amount of data to be processed can be handled well on a single machine - unlike network signaling data

Our approach is to build a modular, scalable and adaptive fraud detection pipeline for the detection of subscription fraud. To achieve this, initially, all the sources of data need to be identified. Mostly, we adhere to Cross Industry Standard Process for Data Mining (CRISP-DM), only the deployment part of the model will not be covered in the scope of this thesis - even though this would be an interesting research topic as well. We start with explorative data analysis to manually search for interesting patterns which could be turned into features for the machine learning algorithms later on. Then algorithms detecting fraud are to be implemented and compared. We will use an individual cost-based approach to deal with imbalanced classes. To better explain the impact of our algorithms to the business we rely on the notion of savings as a representation of money saved by employing the model. As our experience shows, this monetary representation is better understood by managers and controlling department than unfamiliar performance metrics of statistical models. However, we also use classical machine learning approaches and compare both with the outcomes of the current credit check process.

This thesis will be structured as follows:

- *introduction*, giving a brief overview of the problem and proposed methodology
- *state of the art*, defining the types of fraud and explain how fraud is dealt with in other industries and recap of what has been done in telecommunications to prevent fraud, but also give a brief overview of the legal as well as ethical issues involved in predictive fraud detection
- *methodology*, explaining the structure of our dataset as well as algorithms employed in our solution and how we dealt with highly skewed data
- *results*, presenting our results and discussion
- *conclusion*, overall summary and future outlook

In the next section, we will begin outlining different types of fraud.

CHAPTER 2

State of the art

> There is one simple rule to follow in detecting fraud: "Follow the money".
>
> *Richard Becker, (Becker et al., 2010)*

In the next section, we will give an overview about fraud in general, but also particularly about fraud in telecommunications and similar industries. Additionally, we will provide a brief overview of the legal and ethical issues involved with big data predictive fraud analysis.

Fraud means: "wrongful or criminal deception intended to result in financial or personal gain" (Oxford English Dictionary, 2015).

2.1 Hierarchy of fraud

As shown in Figure 2.1, the fraudster can have a variety of relations. Fraud may originate internally from managers, regular employees or externally. For the problem we are tackling, we assume external fraud. External fraud is differentiated into average, criminal and organized profiles. The average offender may just be taking chances and not commit fraud repeatedly. Criminal offenders commit fraud organized and repeatedly, whereas organized fraudsters describe groups of criminals cooperating. To prevent detection professional fraudsters will adapt their modus operandi over time to counterfeit detection systems. As companies interact with a lot of business partners a manual analysis is often infeasible. As this applies to our problem, we will try to tackle it applying advanced computational analytics.

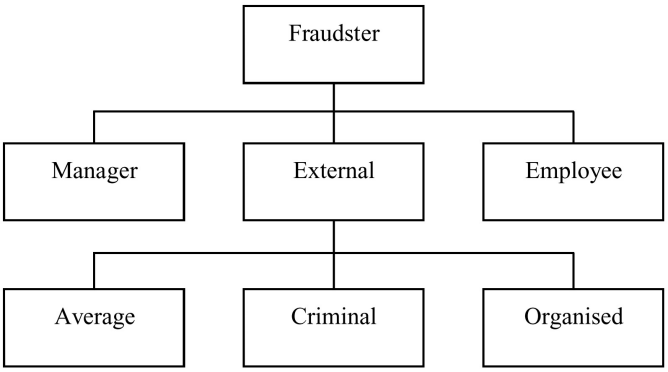


Figure 2.1: Hierarchy of white collar crime (Wang, 2010, Figure 2.1).

2.2 Fraud in the telecommunication sector

Telecommunication provider offers a variety of services with different payment schemes.

2.2.1 Offerings by telecommunication provider

Usually a telecommunication providers services portfolio includes *pre-* and *postpaid* communication services. Additionally, some provider bundle media access or allow for added value payments levied by their regular billing system. As already defined by its name, the prepaid payment scheme assumes money is paid up front before the service can be consumed. Therefore, this product is less prone to be abused by a fraudster.

However, postpaid services are paid for using the service and require trust that the consumer will pay after the consummation of service with the first periode. As one readily can imagine, the potential for fraud is far higher. Speaking of numbers, most consumers pay their premium. For the dataset which was analyzed in this thesis, only 1.9% of the observations are fraudulent. Nowadays most telecommunication providers combine support for financing of mobile devices with postpaid services. This increases the risk for contracts where a pricy smartphone is included "for free." In our analysis, we will focus on postpaid services, limited considering the ones which involve private customers as these show the highest tendency of fraudulent behaviour.

2.2.2 Types of fraud

Moreau et al. (1997) identify six types of fraud:

- subscription fraud
- manipulation of Private Branch Exchange (PBX) entries
- freephone fraud
- premium rate service fraud

- handset theft
- roaming fraud

According to Hoath (1998), subscription fraud is the most common variant of fraud. Subscription fraud is described as obtaining an account and using the telecommunication service without paying for any charges. A differentiation against bad debt, notably for personal usage is hard, especially in the first one to three months of a new contract (Estêvez et al., 2006). However, it can be stated that for subscription fraud the customer did never have the intention to pay, whereas for bad debt force majeure may lead to the customer's default (Geith, 2006; Daskalaki et al., 2003). If fraudsters take over existing legitimate accounts this scheme called superimposed fraud, as a subcategory of subscription fraud and may involve theft of legitimate identities. Often social engineering may be used to accomplish this (Kabari et al., 2015). As we will only deal with new contracts and do not perform differential analysis (Estêvez et al., 2006), we will put a particular focus on trying to prevent any superimposed fraud during account creation through our solution.

Usually, other types of fraud rely on loopholes in technology, but we focus on our analysis on identifying never paying customers to prevent the loss of device and subscription fraud.

2.2.3 Preventing fraud

There is an inherent tendency of the different types of fraud to converge. Thus joint approaches are required to cope with new and combined types of fraudulent behavior (Kabari et al., 2015). A lot of the research contributions focus on using a specific technology to cope with fraud. The ACTS ASPeCT (Advanced Security for Personal Communications Technologies) (Shawe-Taylor, 1999) employ a different solution-driven and combined approach of several models to unite the strength of several models. It was a European joint attempt by several telecommunication providers to fight fraud[1]. Their approach is similar to ours, but we will have a focus on predicting fraud up-front or at least considerably speeding up the dunning process. They use a combination of a rule-based tool for a great explanation why an alarm was triggered, unsupervised neural network to cope with new and yet unseen types of fraud and anomalies and a supervised neural network which has a very high positive prediction rate.

Techniques commonly employed in research projects to prevent fraud are Self Organizing Map (SOM), rule-based data mining techniques, neural networks, decision trees, random forests, Bayesian networks, clustering or mixture models (Estêvez et al., 2006; Geith, 2006; Geith, 2006; Morozov, 2016; Olszewski, 2012; Taniguchi et al., 1998; Xing et al., 2007).

A lot of the approaches deal with fraud after it has happened.

[1] http://cordis.europa.eu/result/rcn/24184_en.html

Post mortem

Historically speaking, simple threshold-based approaches were used initially. As Becker reports for AT&T, they ended up using 30,000 individual thresholds to fine-tune the detection (Becker et al., 2010). Later this led to the development of individualized and signature-based methodologies. Absolute approaches will ring an alarm if a Key Performance Indicator (KPI) exceeds a certain threshold, e.g., two hours per day calling to a foreign country with high interconnections fees.

However, a lot more of the methods to detect fraud in telecommunications deal with differential analysis (Daskalaki et al., 2003; Hilas et al., 2005; Cahill et al., 2004; Subudhi et al., 2015). Another word for it is profiling. It can be performed at different levels. Either single Call Data Records (CDR) are analyzed, or a signature of each specific user is created and compared to new behavior over time (Hilas et al., 2005). Signature-based methods are prevalent and successful. The advantage to absolute approaches is that an optimal threshold for each customer can be defined as the classification is based on the individual call history profile. Differential analysis will only be able to detect fraud after it has happened as pointed out by them.

Wouldn't it be better to prevent fraudulent customers accessing the service in the first place?

Predictive approaches

> Prediction is very difficult,
> especially about the future.
>
> *Niels Bohr*

Optimally fraud could be recognized before it takes place. For a telecommunication provider, this means before a new customer is accepted to their services a just in time screening could prevent future losses. The current solution of conducting a credit scoring agency with this task is not providing results good enough to keep up with rising risk caused by rising prices of fancier smartphones. As shown in Figure 1.1, a considerable time lag has to be noted after a fraudulent account can be closed. Thus it should be a priority to prevent fraud in the first place. As Henecka et al. (2015) conclude, fraudsters might already have identifying profiles at different carriers. However, as data sharing is scarce fraudsters can exploit it. To allow for more data sharing, they propose a unique privacy-preserving distributed data mining approach. We experienced the same problem within the group of our industrial partner. Sharing of data between national subsidiaries is discouraged or often forbidden due to national laws.

Scientific contributions targeting fraudulent new customers in the telecommunications industry are rare. This is not so for other industries like banking: credit scoring has attracted much attention. For telecommunications the following contributions are notable:

Daskalaki et al. (2003) tried to predict customer insolvency. They experimented with discriminant analysis, decision trees, and neural networks and concluded that decision trees offer the highest accuracy for their data set, as well as the lowest error rate.

Estêvez et al. (2006) analyzed data of a telecommunications carrier in Chile. They propose a two-stage model: classification and prediction where the second module predicts the fraudulent potential at the time of subscription. They combine data from the billing process and internal sources with a national insolvency database. A multi-layer perceptron is used to perform the classification.

Our approach will build on gradient boosted trees and incorporate an example dependent cost matrix to take the individual risk for each observation into account.

Summary fraud prevention approaches

For both approaches, it is clear that they have to adapt over time to new behavior of customers. As such they are inherently complex as a new type of fraud may not yet be documented in the training dataset. Both approaches have to deal with highly skewed data: there will be far more legitimate data points than outliers in the database. Possibilities how to deal with such skewed data where non-uniform costs are associated per error was proposed by Estêvez et al. (2006) and Bahnsen, Aouada, et al. (2014).

Fraud is a problem in several other major industries. In the following section, we will give an overview of the approaches developed there with a focus on handling new customers.

2.3 Fraud in other industries

Banking and IT require sophisticated fraud detection technology. In the banking sector, fraud detection technologies are often used for credit scoring and credit card fraud detection. IT is using this technology to power Intrusion Detection / Intrusion Prevention System (IDS/IPS) to recognize advanced threats from hackers and keep them out of a companies networks.

2.3.1 Fraud in banking

Detection of fraud is a diverse problem: credit card transaction fraud, money laundering, telecommunications fraud, computer intrusion and medical & scientific fraud are popular areas of fraud detection (Bolton et al., 2002). We will detail on a couple of them.

West et al. (2016) gives a broad overview about financial fraud in general and serves as an excellent introduction to this topic. They make the point that hybrid methods combining a multitude of traditional approaches strictly targeted at the specific domain are trending. Startups try to commercialize similar big data fraud detection technology by offering improved credit scoring services (A. Lobe, 2016).

At least for banking, it can be said that fraud handling is relatively secretive. Therefore only a few quantitative studies exist (Mählmann, 2010; Dorfleitner et al., 2014;

Hartmann-Wendels et al., 2009). From these, Mählmann (2010) is of special interest to us as they deal with credit scoring. They classify the features most important for a successful credit scoring fraud detection:

- gender: women are less likely to commit fraud but more likely to default
- age: decreasing probability of default until an age of around 70
- nationality: foreigners are more likely to participate in fraudulent activities but less likely to default than locals
- occupation: higher paid white collar workers are less likely to endorse in fraud
- martial status: married people are less likely to commit fraud
- the accounts likely to default cause much greater loss
- high-risk loans face fewer cases of fraud

They conclude, that even though fraud is far less frequent than defaults (insolvencies), it is a lot more cost intensive.

Dorfleitner et al. (2014) denotes that most fraud cases happen in real branch offices 2014, but there is a clear trend towards online fraud. Our industrial partner confirms that most of the fraud they see is conducted online nowadays (T-Mobile Austria, 2016). As Dorfleitner et al. (2014) did not have a variable matching nationality due to anti-discrimination legislation, they needed to emulate it. By comparing non-German names with the names of rejected applications due to the suspicion of fraud and matching it with the customer's Christian name and surname they could obtain a very distinguishing mark of the customer's.

Morozov (2016) is interesting as a one-class Support Vector Machine (SVM) is used to predict whether a credit application profitable as they conclude, that unsupervised fraud detection is possible. Unsupervised algorithms have the advantage that labeled training data sets do not need to be prepared manually.

Similar to telecommunication, credit card fraud can take place during sign-up or via fraudulent transactions (Bhattacharyya et al., 2011). Mahmoudi et al. (2015) use a cost-based learning approach utilizing Fisher discriminant analysis to cope with skewed data for credit card fraud detection.

As fraudsters often change their tactics the detection system is required to evolve as well (Correa Bahnsen, Aouada, et al., 2016). They propose a feature engineering and selection approach which leads to an average increase in savings of 13% by creating aggregations of credit card transaction at different time intervals.

Van Vlasselaer et al. (2015) propose a new fraud detection methodology based on intrinsic features and structure derived from the graph network of credit card transactions using random forests, logistic regression and neural networks for classification.

Dheepa et al. (2013) use modeling of animal behavior and SVM to model the fraudulent behavior of cooperating groups.

2.3.2 Fraud in IT

As more and more data and business processes are put into IT systems the interest of hackers to attack these has risen. IDS/IPS is a standard device deployed to recognize attacks and prevent them.

To not only rely on signatures but also be able to detect new and advanced threat models increasingly machine learning is used (Vrat et al., 2015; Pernul et al., 2015; García-Teodoro et al., 2009) to detect anomalies by applying logistic regression, decision trees, and random forests. As hackers are creative and no advanced threat will be similar to known patterns (Patcha et al., 2007) often anomaly detection algorithms are employed.

Contrary to the Banking sector, metron[2] and ids-hogzilla[3] are great examples of open source software available to combat threat induced by hackers.

In the next section, we will provide a brief overview of legal and ethical problems for big data fraud mining and prediction.

2.4 Legal implications & ethical problems

> We kill people based on metadata.
>
> *General Michael Hayden, former director of the NSA and the CIA*

Before big data analytical capabilities were widespread, usually, fraud could only be detected after it had taken place or after the crime had been committed. Our problem is very similar to predictive policing where people may be detained based only on suspicion (Carter et al., 2009; Ferguson, 2012). General Michael Hayden, former director of the NSA and the CIA, confirmed that these secret services even go for the next step and kill people on the grounds of metadata predicting their suspicion.

2.4.1 Legal aspects

The Austrian Federal Communications Act 2003 (TKG) specifies that:

- §25a (1) telecommunication providers need to offer a tool for customers to control their cost and protect them. (2) A suspension of service free of cost in case of

[2] `https://metron.incubator.apache.org/`
[3] `http://ids-hogzilla.org/`

excessive usage applies to prevent the customer from too high cost. Due to this law, the national regulatory authorithy is empowered to establish a regulation. This is the so called Kostenbeschränkungsverordnung. It defines, that every customer may use data up to an amount of 60 EUR per month. After reaching this limit, the provider has to cut off the data service unless the customer agrees to go on using data and is willing to bear the costs. On the Europen Level there is the so called European Romaming regulation which defines that all over the world this 60 EUR limit for all services has to be established (also including voice and SMS).

- §70 cancellation of service is only applicable after repeated reminders. Emergency calls must not be blocked (see 1).
- §97 (1) Master Data is very strictly regulated. These data about the customer may be used to change or terminate a contract, for billing reasons, for the telephone-book and for request of emergency agencies.but the data needs to be deleted or anonymized latest 6 month after the termination, as long as no other legal obligation exists (e.g. Tax-Law) or the customer has given a freely consent to keep the data stored for a longer periode.
- §99 (2) traffic data needs to be deleted as soon as possible. Storage needs to have a special purpose e.g. to calculate billing information. When the bill is paid and this information is no longer required the data needs to be deleted or anonymized. Traffic data with billing relevance needs to be stored in Austria for a maximum duration of 6 month to proof the correctness of the bill towards the customer. This obligation is defined in the regulation of the regulatory authority, the so called Einzelentgeltnachweisverordnung) together with §99 TKG 2003.
- §99 (2)2. in case the bill is not paid after a reminder the data does not need to be deleted.

As the Austrian regulations are relatively similar to Germany in this regard the German TKG does not propose significant differences in regulation. It specifies that by §45k an account may not be canceled due to minor default. At least 75 € is required.

Additionally, privacy regulation rules apply according to the Austrian Datenschutzgesetz (DSG).

No reference regarding predictive suspicion is found in the TKG. In general, only data required for regular operation of the business is allowed to be stored. However, anti-fraud activities and the data required for these are rather tensile which leads to many use cases being possible were considered illegitimate otherwise. We strictly adhere to these official but also company internal privacy regulations during this analysis.

2.4.2 Ethical problems

For the research community, the analysis of public datasets was never seen as a problem. Hauge et al. (2016) used *only public* datasets to carry out geographical profiling which

is typically used only for murders or serial crimes. After combining several big data sets further insight which was not present in the pay of the raw data sources can be gained. Metcalf et al. (2016) wish that more researchers would adhere to critical data studies which critically reflect the impact of *data subjectivity*.

These issues are of high relevance in this particular field as a multitude of telecommunication use cases are developed to analyze network signaling data at varying levels of aggregation[4].

2.4.3 Summary legal & ethics

As one can see, these problems are complex and must not be underestimated. This is especially important for a company which has realized that value can be generated out of their massive pile of data by applying predictive analytics to more and new business cases which often means sharing the data to some part with external partners. In the face of the new GDPR regulation, this topic, in particular, shows its importance.

We will try to be very careful with data protection, e.g., some peculiar features will not be analyzed in plain but only as hashed features and apply the local regulation as well as internal company policies to protect personal information.

[4] https://thehftguy.com/2017/07/19/what-does-it-really-take-to-track-100-million-cell-phones/

CHAPTER 3

Methodology

Garbage in, garbage out.

An IT wisdom

3.1 Data mining process

Multiple standards are common for data mining (KDD, SEMMA, CRISP-DM). According to Chapman et al. (2000) CRISP-DM is the most widely accepted standard. As CRISP-DM is very common in the industry this thesis will adopt it for the analysis process. Figure 3.1 depicts the concepts involved in CRISP-DM. However, we will only partially adapt methods from the industry standard as some of the proposed concepts may be out of the scope of a thesis. This especially applies for *Deployment*.

As Liu (2016) point out, in a real-world scenario CRISP-DM will need to be accompanied by analytical approaches selection, results explanation and feedback loops for complex machine learning projects. They propose a detailed step-by-step approach which may help to develop a causal model (Liu, 2005).

This chapter will focus on the following steps:

- *data mining process* introduction to CRISP-DM
- *business understanding*, requirements analysis and macro overview
- *data understanding*, data sources, data collection, getting familiar with the data
- *data preparation*, data cleaning and feature engineering

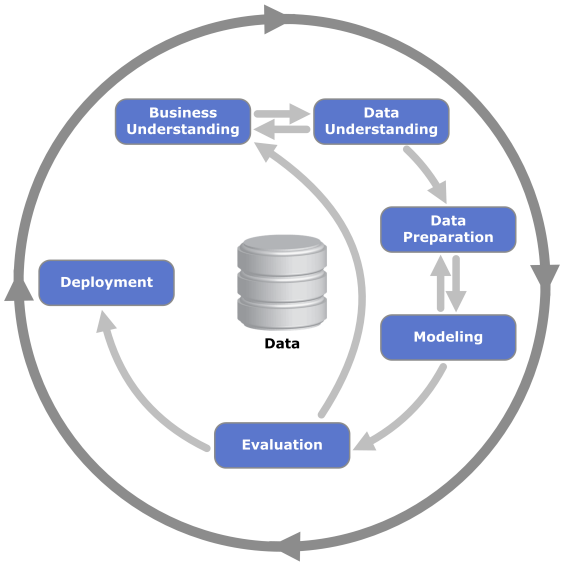


Figure 3.1: CRISP-DM Data mining process (Jensen K., 2012) originally proposed by Chapman et al. (2000).

- *dealing with highly skewed data* outlining several strategies how to cope with such datasets
- *models* define the mathematical formulations of the different machine learning models
- *model optimization* will outline how we applied hyper parameter optimization
- *cost matrix formulation for telecommunication industry* defines the mathematical reasoning behind our proposed cost matrix

3.2 Business understanding

Currently, creditworthiness of new customers is evaluated through an external credit scoring provider. This score relies on past facts, e.g. unpaid bills (T-Mobile Austria, 2016). Such a basic check is catching parts of the problematic customers. However, it is optimized to detect credit default and not *neverpayers*. The result is a traffic light like warning system which is steering business processes. Customers marked as yellow require further manual checkup from our partner. Many customers are classified as the manual credit check process by the current system which leads to a considerable amount of work for the manual credit check process. Additionally, the *green* segment of customers holds a remarkable number of neverpayers as well - unfortunately, these are accepted automatically.

Even for the credit scoring agency, it is impossible to have already collected enough data for each new customer. Some of these are marked as trustworthy. Others are required to hand in further identifying documents by following the manual credit check process.

Our industrial partner confirmed that even though these systems are in place, several million € of uncollectible debt needed to be declared in increasing loss during the last couple of years. This translates up to 400000 € per month. Individual losses vary depending on the device, length of dunning process and costly roaming interconnect fees. As shown in Figure 1.1 on page 3, the biggest initial loss is the price for high valued smart phones. Our partner observed that usually, fraudsters choose expensive phones with a high resale value, e.g., *iPhones*. Secondly, it was observed that fraudsters target expensive contracts where these pricey phones are included for only small or zero fees (T-Mobile Austria, 2016).

From a business's perspective, it is clear that these losses need to be reduced. However, telephone services are considered essential services in a digital society, and it is not possible to terminate such a contract immediately, let alone due to some vague predictions of fraud, according to Austrian TKG §70.

Due to these legal implications, the standard dunning process takes around 90 to 120 days. Internal research has shown, that there is room for improvement while still staying within the legal boundaries (T-Mobile Austria, 2016) when applying advanced analytical algorithms. From a legal perspective, it is essential that at least a single bill was not paid.

Goals of the business are to:

- catch as many fraudsters as possible and even accept a limited number of false positives to reduce the overall cost associated with fraud. At least 50% of neverpayers per month must be identified
- sell a contract even if some algorithms suggest a fraudster, but increase the initial down payment to limit the loss and further scare off fraudsters
- recognize fraudsters as early as possible, at best directly at the POS
- monitor initial behavior to decrease credit limits if a customer is predicted not to pay the bills to reduce losses and additionally trigger an early termination to speed up the dunning process
- adhere to current legal and ethical standards but modernizing business processes with smart AI & machine learning based solutions

3.2.1 Data acquisition

Similar to Shawe-Taylor (1999), billing and monitoring data is an important source for our analysis. On the one hand, the data is collected during the sign-up request process

of a new customer and on the other, data is collected in a variety of internal systems to track usage and check the validity of initially supplied data. The data is enriched via external scoring providers. This thesis, however, can only use data available at T_0 directly at the POS as no usage data can be monitored before the phone or contract is sold.

After a while, the data arrives in our partner's *hadoop*-based data lake and Data Ware House (DWH). It has to be noted that for timely fraud detection some architectural changes are required to receive the required data from new customer contract applications in near real-time due to the legacy architecture of the Customer Relationship Management System (CRM) system which currently drives most business processes.

3.2.2 Makro overview

Profound understanding of the data from a business's point of view is critical for efficient fraud detection. Often data is stored in silos for each department. One part of this thesis is to identify all available data sources.

The following types of data sources were identified. Not all are useful at T_0, and some require a large up-front investment to make them usable for analysis:

- **CRM** *basic data about the customer, contract, hardware, dealer for activated contracts.* Data which is collected during the sign-up process is stored here once (post activation). It contains fields for legitimation, name, age, address, legitimation id about the customer. Device and contract which are activated when the credit check is positive are stored here as well. Additionally, information about the dealer is referenced with the sales transaction. This data source is used for this thesis.

- **Pre-CRM** *basic information for not yet activated customers.* Before the credit check is completed the basic data is stored here and shared with external credit agencies. The data is not updated, but only transferred to the regular CRM when a contract is activated. It contains the history of pre-activation information.

- **DWH** *analytical data storage.* The DWH contains information from various systems. Often enriched to allow for easier analysis. However, the data is not available in real time. Depending on the type of data batch updates are applied once a day to once a month. Information is historized and kept up to date when data changes are recorded.

  There exists a new and old variant of the DWH, data from both systems are used in the thesis.

- **Billing system** *direct sales transactions.* In the local stores of our partner, the billing system is tracking information about sold devices as well as additional equipment and services. It is only interfacing with the SAP. Unfortunately, sales transactions by the same customer are not necessarily referenceable to the same

customer. I.e., insurance is resold from a financial services provider, but it is hard to find out if a customer who is getting a new contract is buying insurance with the new device to protect it. Therefore this data source is not used in the thesis.

- **SAP** *controlling & financial KPI.* SAP is storing financial information; i.e. pending payments can be found here. In general, there is no direct access to this database. One needs to apply for interfaces. Partial data from this data sources are used in the thesis.

- **Web shop** *clickstream, digital fingerprint.* The online shop is gradually generating more sales events than regular stores. Online, a lot more metrics collection possibilities exist. Our partner is tracking clickstream information and a digital fingerprint. This information is not used in the thesis due to not enough historical data being available.

- **Usage data stream** *from network signaling data.* Network signaling data constitutes the most significant amount of data which is continuously monitored but only available when a mobile device is used, i.e., not available at T_0. It can be consumed on several aggregation levels. The raw data stream offers tremendous analytical possibilities and is explored in different use cases but is not used in this thesis.

- **CDR** *call records used for billing purposes.* are generated from other systems. They are less fine granular but more accessible as a lot less of information needs to be processed. This data is not used in the thesis.

- **Fraud database** *all types of fraud historized.* The department of customer finance is storing fraudulent customers in an access database for a longer period. Mostly, fields are similar to CRM, but additional information from courts is stored there as well. Also, not only defaults or neverpayers but any type of fraud is stored there. This data is not used in the thesis.

- **External credit scoring agency** *current scoring methodology.* The credit scoring partner is currently providing two metrics:

  - score: a value range from 350 - 700, larger is better
  - decision: traffic light system which is driving business processes. *Red*: customers are automatically disabled, *green*: automatically accepted. *Yellow*: manual processing required.

  The number of yellow customers is fairly large. In the future, an extended fraud score which incorporates the digital fingerprint should be provided as well. Already available data is used for the thesis.

- **Binary files** *images referenced in the CRM.* For yellow customers, the manual credit check process requests documents which constitute a proof of residence or

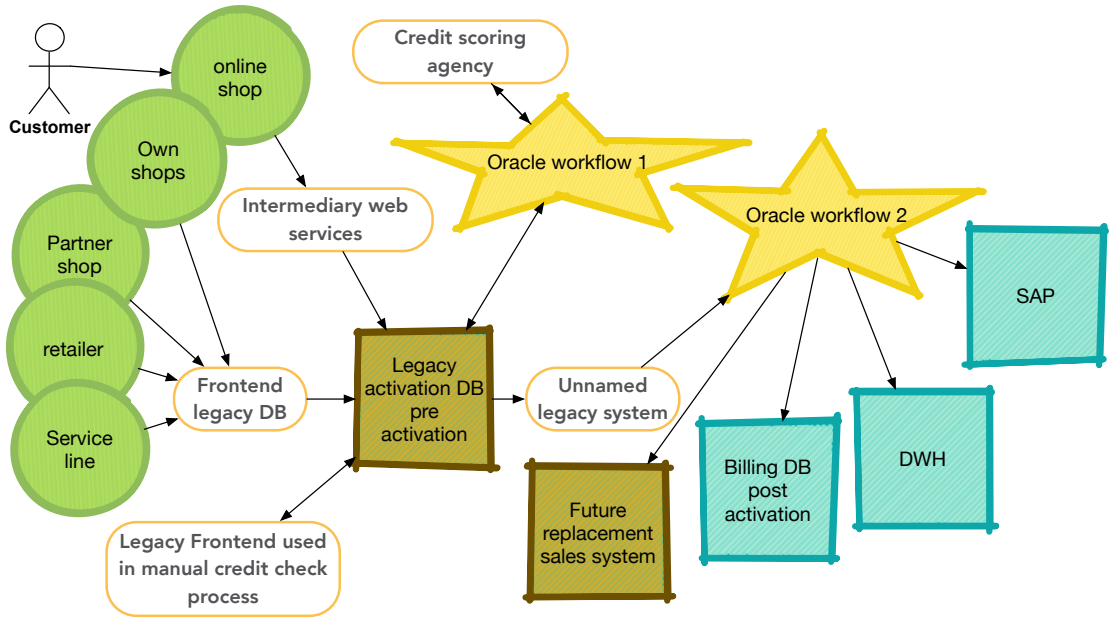


Figure 3.2: Information flow for a new contract through IT systems.

identity. These documents are not stored in the CRM directly, rather only a file path as the reference to a network drive location is saved. Additionally, copies of the signed contracts are referenced similarly for all customers. Not used in the thesis.

- **Credit scoring data from other subsidiaries** *data sharing with subsidiaries.* Our partner belongs to a large group of telecommunication providers. Other subsidiaries hold credit scoring information as well which could be valuable if a customer with a bad credit history is moving to another country. Currently, this data is not shared cross subsidiaries and therefore not available for this thesis.

In general, data quality is a huge problem with such a variety of data sources, but in particular with the older legacy databases. Initially, we were extracting data which was available in an easy to use the structure in the DWH. Later, started to switch to the source systems, i.e., the legacy activation system for reasons of quicker data availability and access to the original data records.

Activation process overview

To build a good fraud classifier especially non-technical, but a business-minded understanding of the data is important (Morozov, 2016). We outline the flow of information through the IT systems in Figure 3.2.

A sales transaction is initiated from a shop, retailer, service line or the online shop. The physical shops use a frontend application which directly writes to the legacy database

which in return holds all necessary data records before a successful activation of a contract is completed. The online shop is using an intermediary web service to access this database. A combination of Oracle workflows is issuing web requests to interact with the external credit scoring agency. In case of a positive scoring result, the contract is activated and stored in a second post activation legacy database. Additionally, subsystems like the future billing system, DWH and SAP are notified by the workflow.

The scoring result is translated to a traffic light system and partially processed automatically. In case of *yellow*, the application is transitioned to a manual credit check workflow. Usually, identifying documents are requested and validated. If these provide sufficient credibility of identity and good credit status customers are activated manually.

3.3 Data understanding

In the following section, we will give an overview of selected features available in the data. Aspects like visual summaries as well as the quality of the columns are addressed per group of features.

When selecting features, we are starting out with the hypothesis proposed by Mählmann (2010) which line out some useful features in the banking industry. However, as the telecommunication industry does not use such stringent Know Your Customer (KYC) processes some of the suggested features are not useful in our context as data is not well maintained and quality suffers too much.

For many features visually there is only a little difference between the distributions of regular and fraudulent users. The power of the features comes from having many mediocre discriminators where patterns are only visible with advanced classification algorithms.

Note, the plots in the next section will only show records where the fraudulence score is > 0 and more than at least 25 fraudulent observations per group are available.

Even though the quality is not optimal for all features, even fields which contain many NaN values can turn out to be an essential discriminator in some cases. One rather needs to think which fields identify regular customers to find good discriminating features which might be the case for a field which lacks a lot of values.

3.3.1 Target variable

Our partner has collected ground truth data for about two years worth of data, which contains a binary target variable called *NEVERPAYER*. All records which contain 1 in this field are to be considered as fraudulent customers. It contains about 400000 records and is massively imbalanced. Only 1.9% of all records represent a fraudulent customer.

See Section 3.5 for details how to handle such datasets.

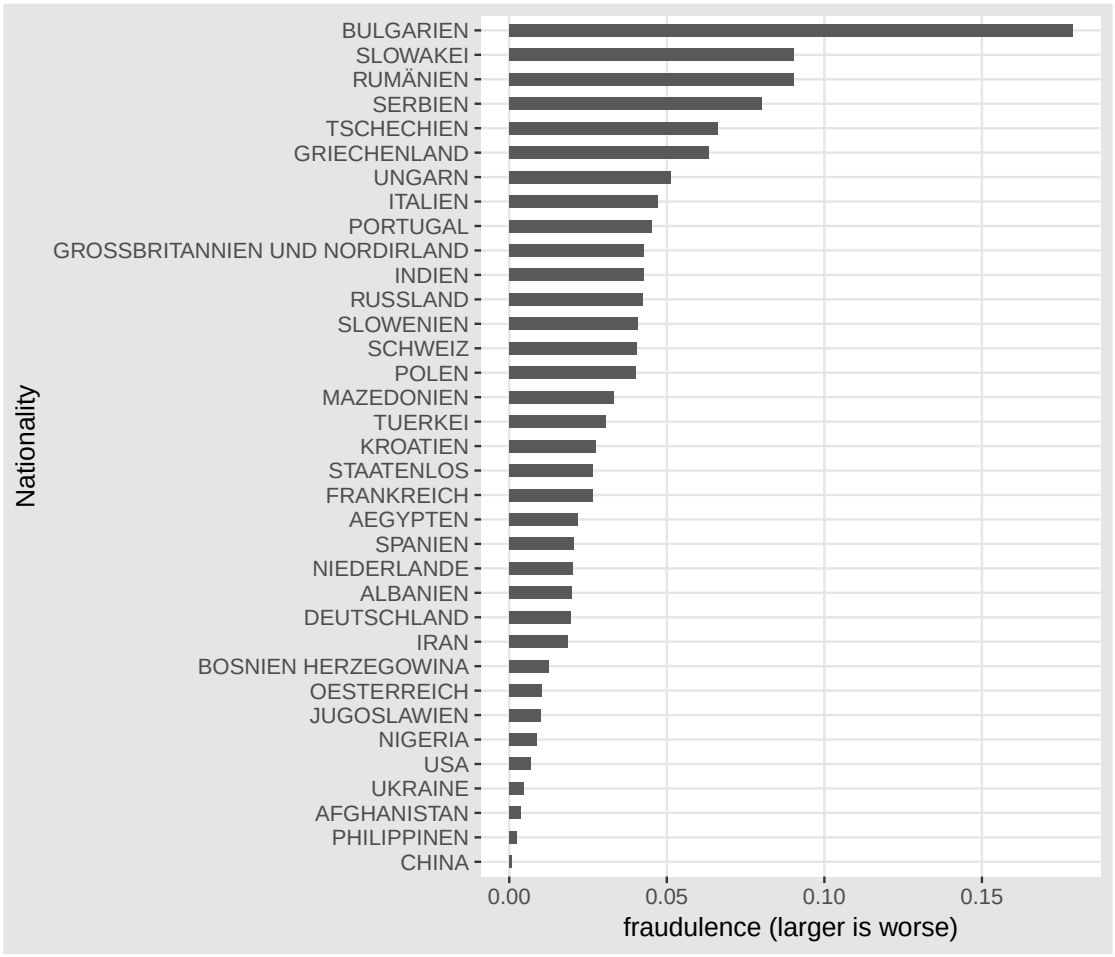


Figure 3.3: Proportion of neverpayers per top 35 nationalities

3.3.2 Features about the customer

In general, these are some pretty standard features like name, age, address. In some countries, nationality is forbidden to be used. Not so in Austria. However, creative researchers found a solution to this problem by inferring the nationality from the sound of the name (Dorfleitner et al., 2014). The data quality for this feature is good. As noted by Dorfleitner et al. (2014) and others the nationality of a customer is a good predictor, especially if it does not equal the local nationality. Figure 3.3 can show that certain eastern nationalities may pose a higher risk customer, but one must not conclude that in general, these are always fraudulent. Interestingly, Germans show a higher fraudulence score than Austrians. The customer finance department has observed multiple cases where Austrian credit scoring agencies did not have any adverse observations for prospective German customers that when asking German ones these people had a bad credit history there in the other country. Unfortunately, this helpful information is usually not available when deciding the credit check process. Also, the local nationality

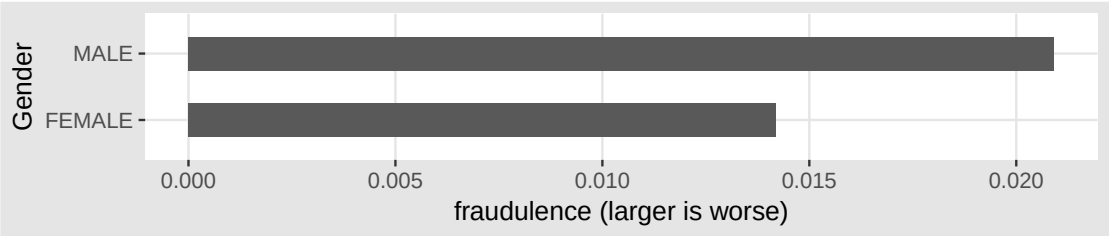


Figure 3.4: Proportion of neverpayers per gender

(Austrians) apparently pose a higher risk than some foreign nationalities. This can be attributed to the imbalance of records in the data as the local nationality is obviously overrepresented.

Only a few customers use their academic title when signing up for a mobile phone contract. This is particularly interesting, as people in Austria usually are very keen on using their academic titles. No customer with a real academic title turns out to be fraudulent. For this feature, the quality is good enough to be useful. In our data, the feature gender shows a somewhat similar distribution which reflects what is reported by Dorfleitner et al. (2014), see Figure 3.4: male customers pose a higher risk for our partner. One has to note though, that gender contains a value *F* which resembles small companies which are treated as a regular private customer and not like a hierarchical business customer and *S* stands for public institutions or clubs, but both of these do not pose a fraud risk high enough to show up in the plots. Again, the quality of the data is ok.

In our dataset the feature age did not turn out to be a good discriminator to identify fraud see Figure 3.5 which shows that both distributions are almost identical. For regular customers, a small bump at around 110 years can be seen. This denotes an outlier in the data as all values of age greater than 110 are limited to this value.

Address for the contract which is not Austrian poses a higher risk. Figure 3.6 displays the country for the contract address and the fraudulence calculated as the percentage for each group compared with the neverpayers per group. For this plot, the minimal number of observations was set to 4 as almost all contract addresses are obviously local, i.e., Austrian. For an Austrian telecommunications carrier, a contract usually requires an Austrian address of residency.

Our partner is offering multiple brands on the market. *T-Mobile* refers to private customers of their prime brand which focuses on offering high quality but pricier contracts and devices. Telering again refers to private customers, but these only are offered cheaper handsets. As Figure 3.7 shows, cheaper handsets result in drastically less risk for the whole brand. Additionally, our partner is providing mobile network coverage for several other brands as Mobile Virtual Network Operator (MVNO) but as fraud, there has no direct impact on our partner's financials this is not considered here.

Visually it is hard to tell from the histogram of Figure 3.8 if the number of contracts

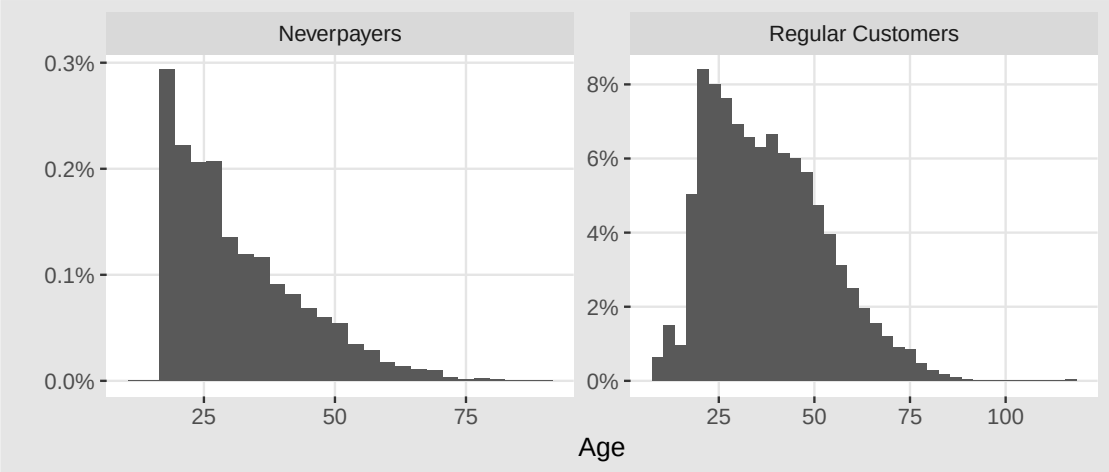


Figure 3.5: Fraudulence per age

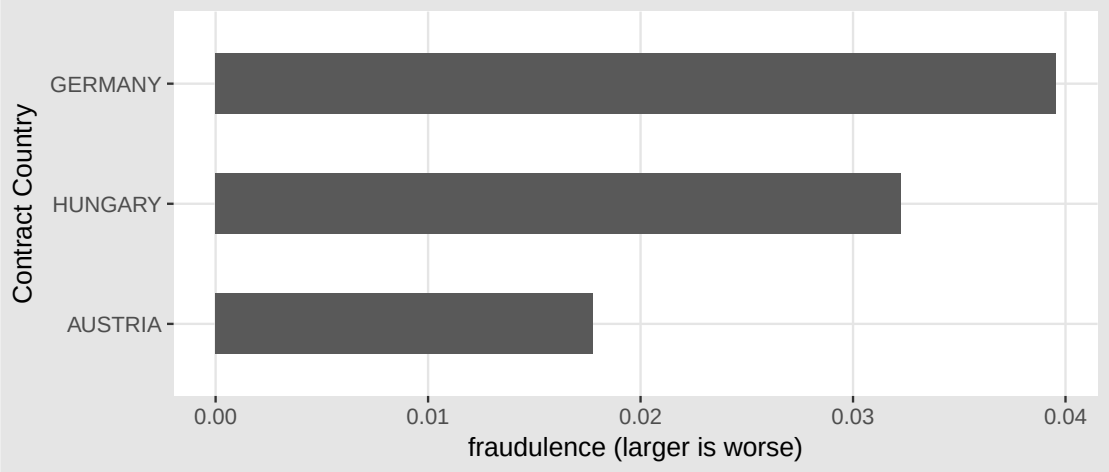


Figure 3.6: Proportion of neverpayers per country of contract address

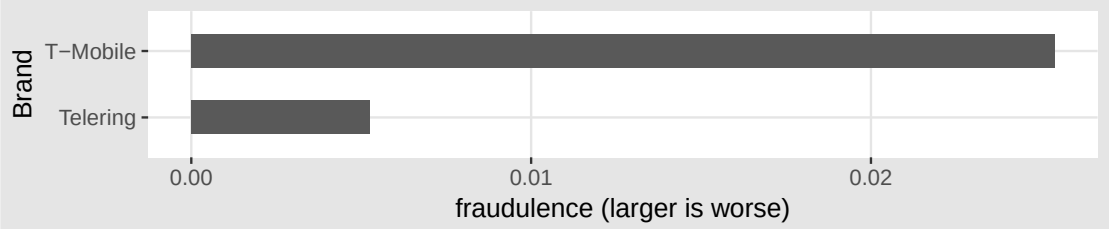


Figure 3.7: Proportion of neverpayers per brand

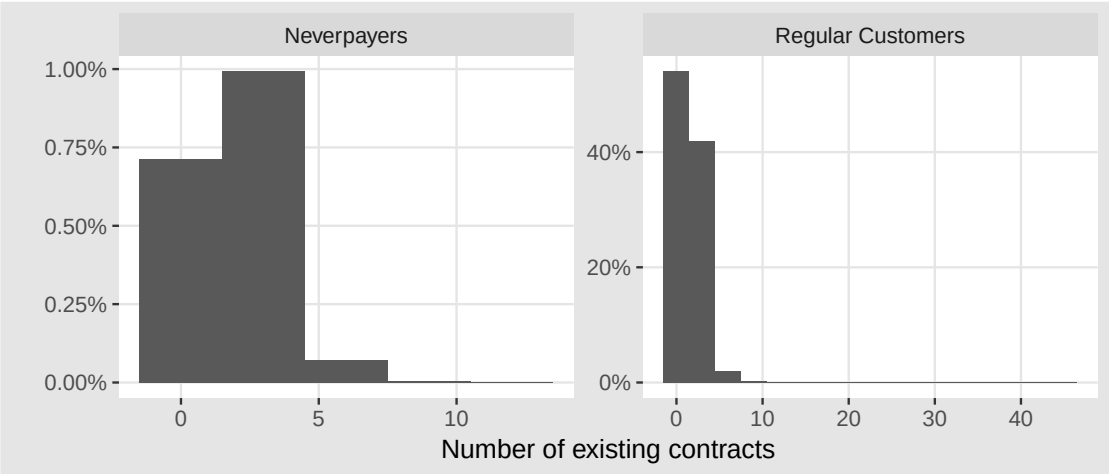


Figure 3.8: Fraudulence by number of contracts

is a good discriminator. One can see that neverpayers tend to have a little bit more contracts than regular customers. As a neverpayer is defined as a new customer where no prior contract existed this feature should not be too helpful as all new customers would have 0 previous contracts. However, fraud often involves identity theft. Identities from public organizations or for example a local doctor are sometimes abused to sign up. It can help to identify these cases.

The ID document constitutes a proxy variable for nationality and seems to be reasonably helpful. In contrast to Germany, in Austria, an ID card is not mandatory. That is why Austrians often use the driver's license as an official document when an identification is required. Instead of foreign customers usually, use their passport when identification is required. Figure 3.9 shows fraudulence per type of identification. Unfortunately, this feature is not available for several observations. Interestingly, there are a lot of fraudulent observations among these data points.

As mentioned before, the credit score is translated to a traffic light like system. Customers marked as *yellow* pose the highest risk out of these three values according to the data. Interestingly, as Figure 3.10 shows, the fraudulence score for unknown values is the highest. A possible explanation is that a contract continuation or the application for a second handset is getting too much trust in advance as no additional credit check from external scoring agencies is requested. Interestingly even some *red* customers turn out to be neverpayers. These are special cases of this type of customers usually are declined automatically. However, in these cases, this decision must have been overruled manually. The credit score as shown in Figure 3.11 is of good quality if requested and a rather helpful feature. However, using it as the sole discriminator, it is not good enough. It is by far more helpful than the traffic light system to differentiate between regular and fraudulent customers, and the business should consider changing the bins of the traffic light system to improve the current scoring result. Interestingly, over time the score

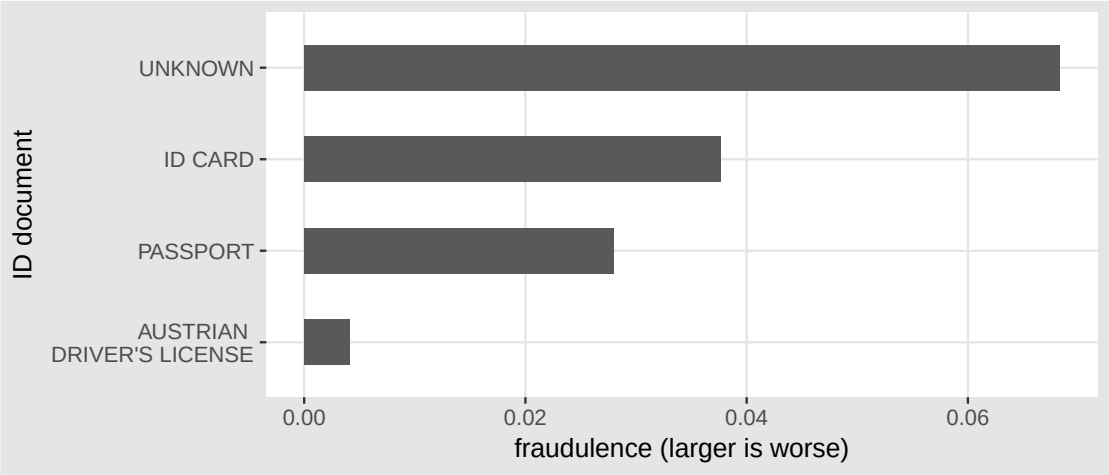


Figure 3.9: Proportion of neverpayers per ID document category

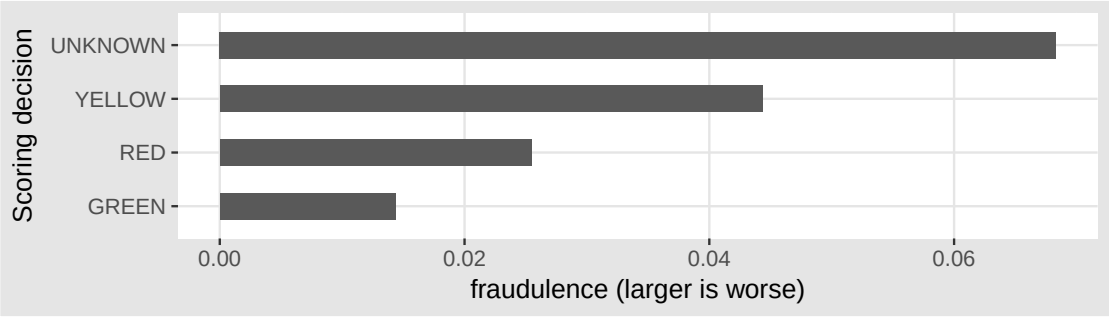


Figure 3.10: Credit scoring decision

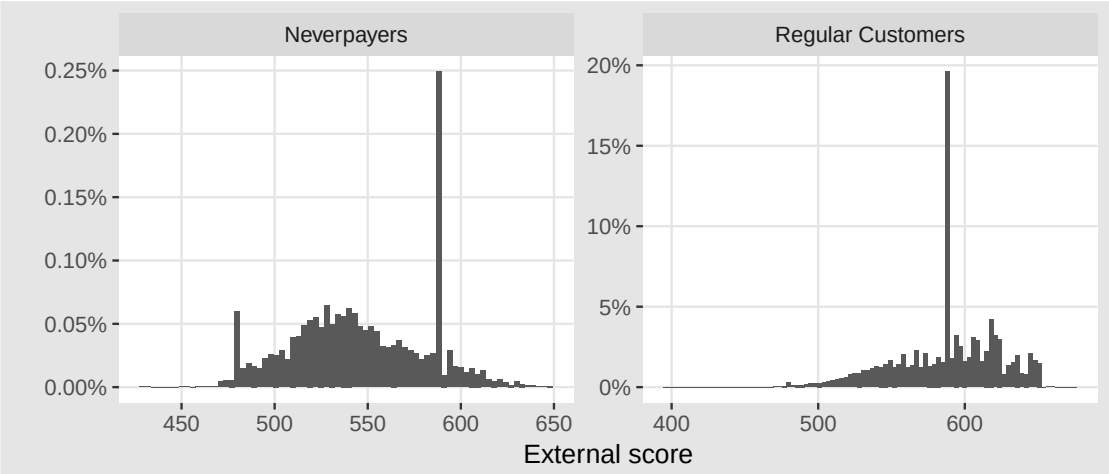


Figure 3.11: Fraudulence by scoring results of external credit scoring agency

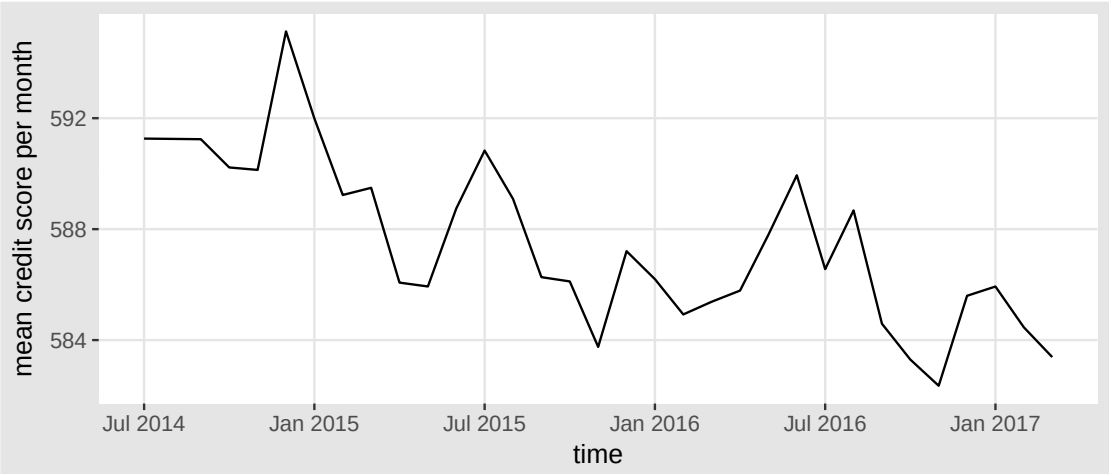


Figure 3.12: Scoring results over time

changed quite a lot. Figure 3.12 shows that there must have been a rule change in the last year to explain the drop in numbers.

3.3.3 Features about the dealer

Like for a customer address, company and seller name are recorded in the data. However, the quality of the data is not as good as for customers. Some stores are located in highly frequented regions. Others are placed in a shopping mall or near a public transport location. The location can have a high impact in the overall fraudulence per shop as Figure 3.13 shows that the lighter blue shops with more customers turn out to have a higher fraudulence score. Interestingly, these are located in a shopping center in Vienna which is not located in the best area of the city. The color symbolizes how many neverpayers were found in stores. Nearly all stores listed in this hit list are fairly small retailers with the exception of a big chain and two of our partner's stores.

We initially expected that online sales contain the highest fraud percentage as most fraud attempts occur online (T-Mobile Austria, 2016). However, the data falsifies this. Inquiries show that additional manual checks are in place which turns out to be highly efficient as Figure 3.14 depicts. Data quality for this field is not optimal. Also, a similar but slightly different field is contained in the dataset.

3.3.4 Features about the device

Several metrics are collected about the device. Usually, they are financial KPI. Some were introduced relatively recently and are not available for all observations, but the quality of the data is still good enough to be useful. A lot of different but efficiently similar metrics exist when considering the price of the device. The difference depends on the cost applied.

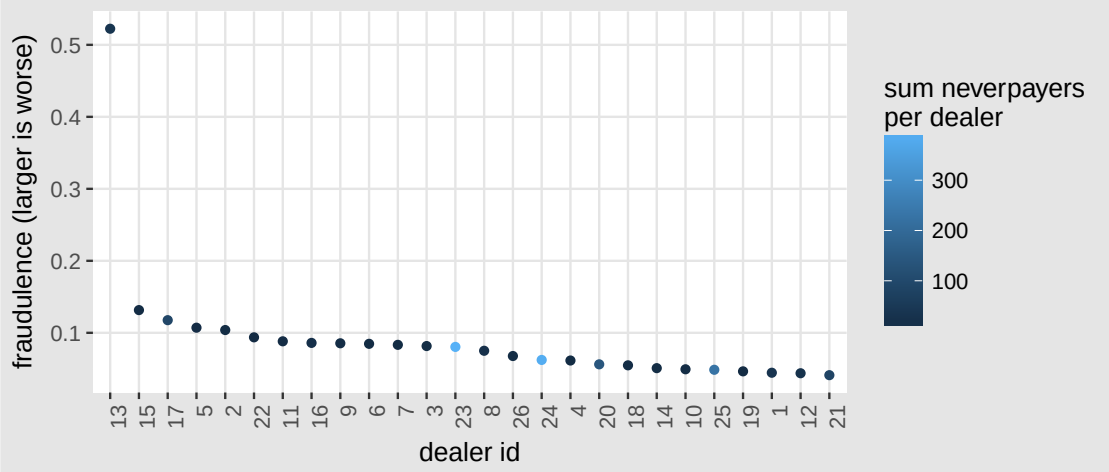


Figure 3.13: Most problematic dealers with at least fraudulence score of 0.04 and at least 25 neverpayers

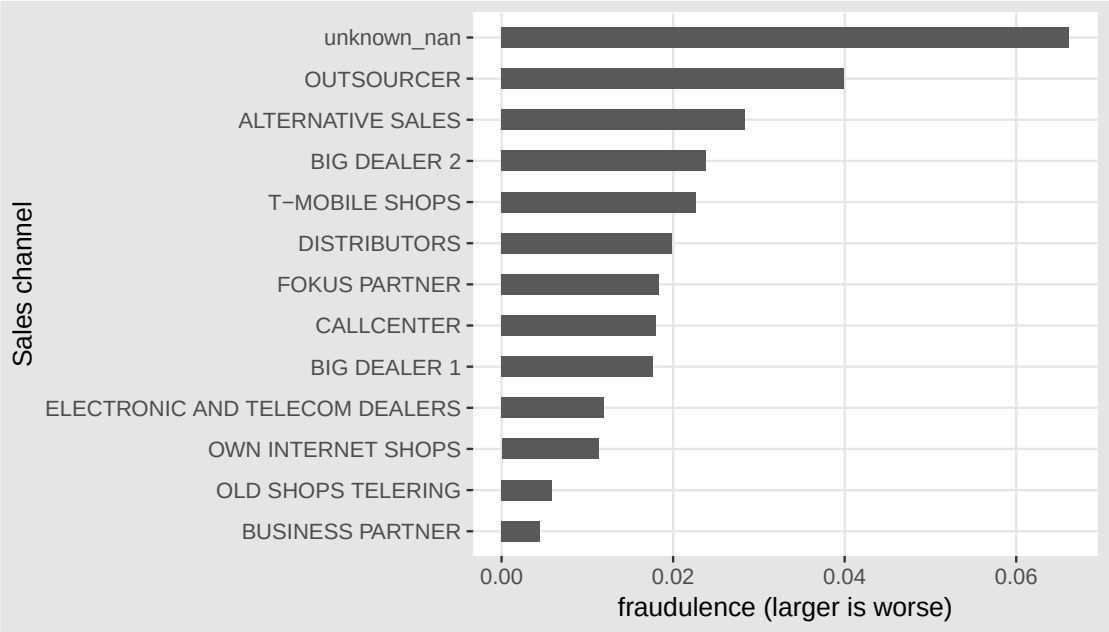


Figure 3.14: Fraudulence per Sales channel

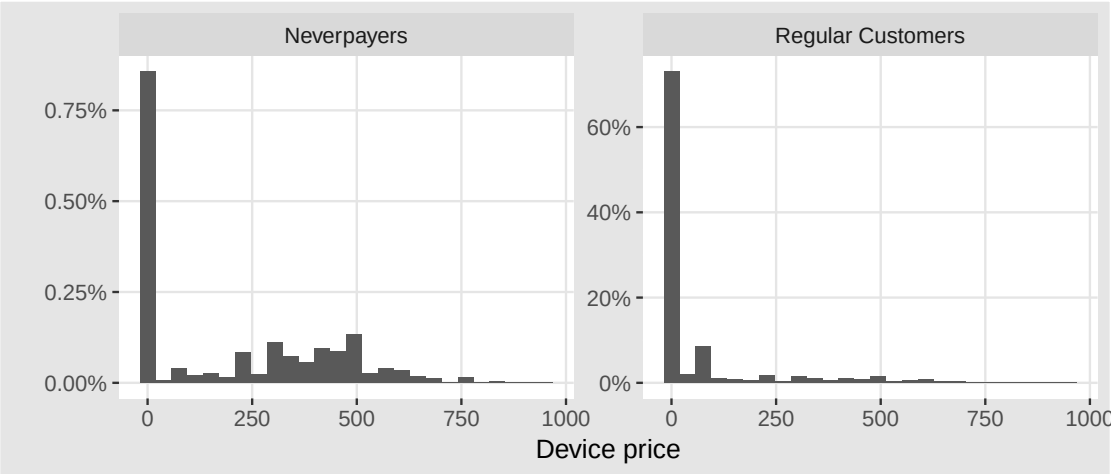


Figure 3.15: Fraudulence per device price without subsidy

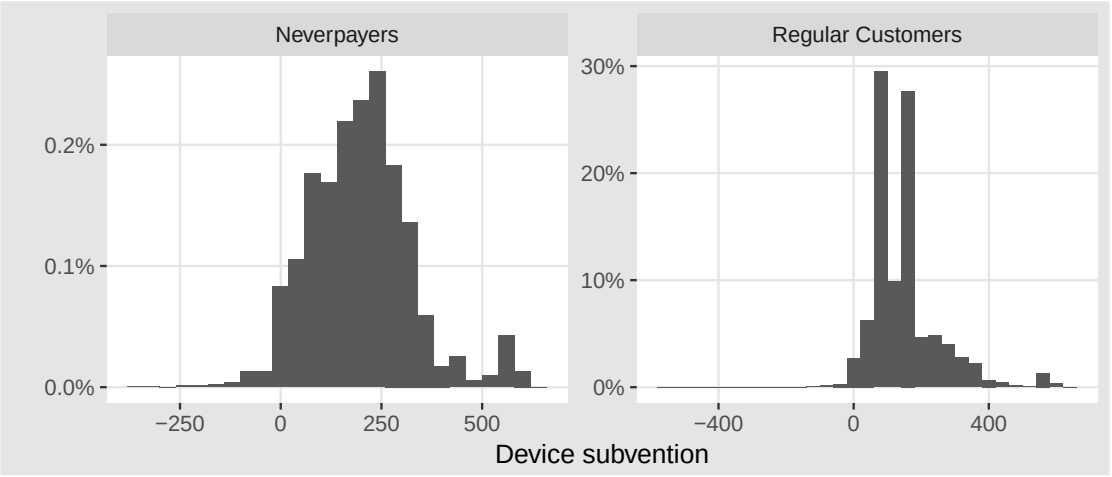


Figure 3.16: Fraudulence per device subsidy

Figure 3.15 depicts the price of the device when the subsidy for it by our partner is already deducted. One can see that there are a lot of very cheap devices. Similar to normal customers, neverpayers prefer these costly devices which often are offered in premium contracts for 0€ or 1€ as the subvention as shown in Figure 3.16 often is very high. For neverpayers, it looks like it is a little bit higher than for regular users, but a visual discrimination between regular customers and neverpayers solely using this feature is hard.

The real price for the hardware which is at risk, i.e., where no subsidies are deducted is visualized in Figure 3.17. There are two peaks for medium priced devices (around 250€) and high-priced devices (around 600€). For regular customers, the medium priced devices sell more often than for neverpayers. Though in absolute numbers regular

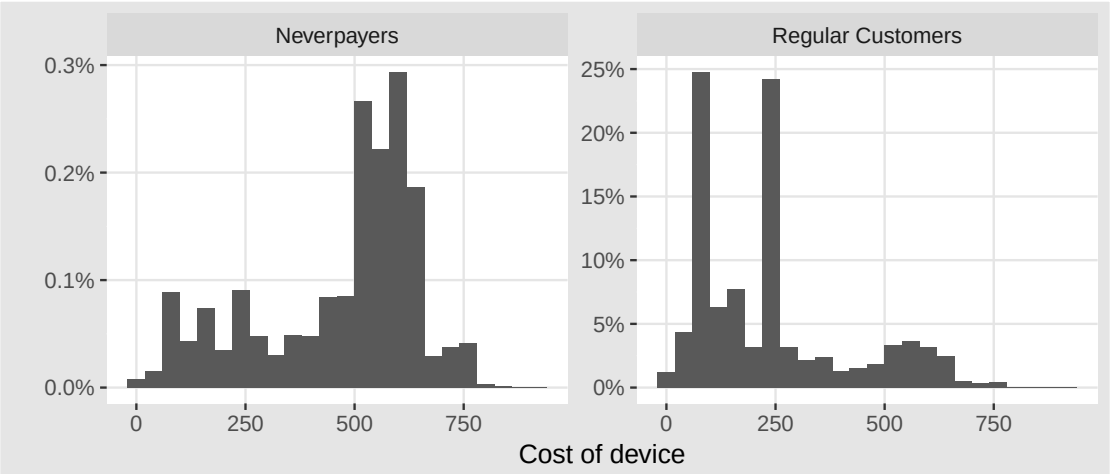


Figure 3.17: Fraudulence per device price

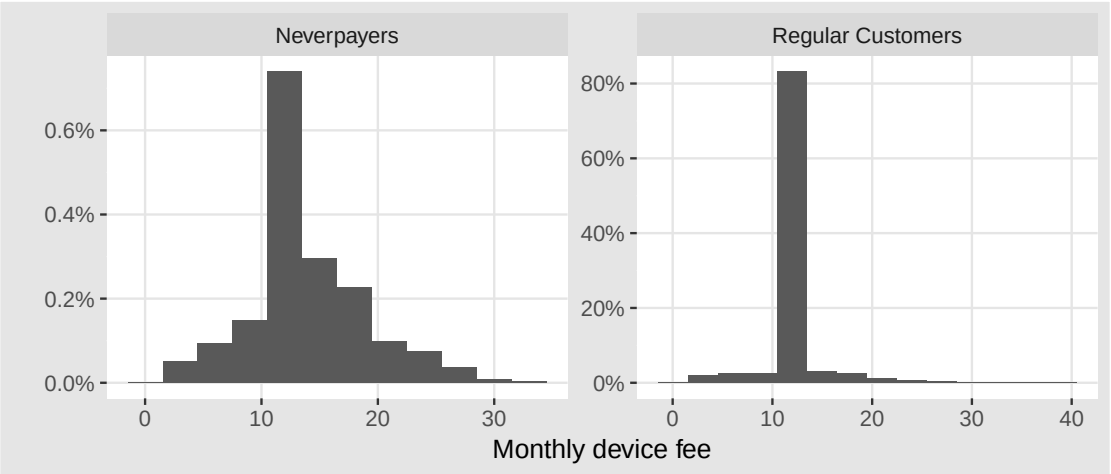


Figure 3.18: Fraudulence per monthly device hardware rate in €

customers buy more expensive devices than neverpayers exist.

The monthly cost for the device and usage are referenced separately. Figure 3.18 shows again a visual discrimination is hard as the absolute counts cant easily be compared here.

High priced smartphones, i.e., iPhones from Apple do not only mean that our partner is losing a lot, but as Figure 3.19 shows these, in particular, are often involved in fraud cases. Possibly this is because these devices offer a high retail value.

Looking at the on-device storage and color (Figure 3.20, 3.21) it is apparent that devices with yellow color and a large amount of storage pose the highest risk. As seen in Figure 3.19 devices from Apple are leading the fraud score. Large devices and a greater variety of colors (yellow, red) were only introduced relatively recently. Therefore, we conclude

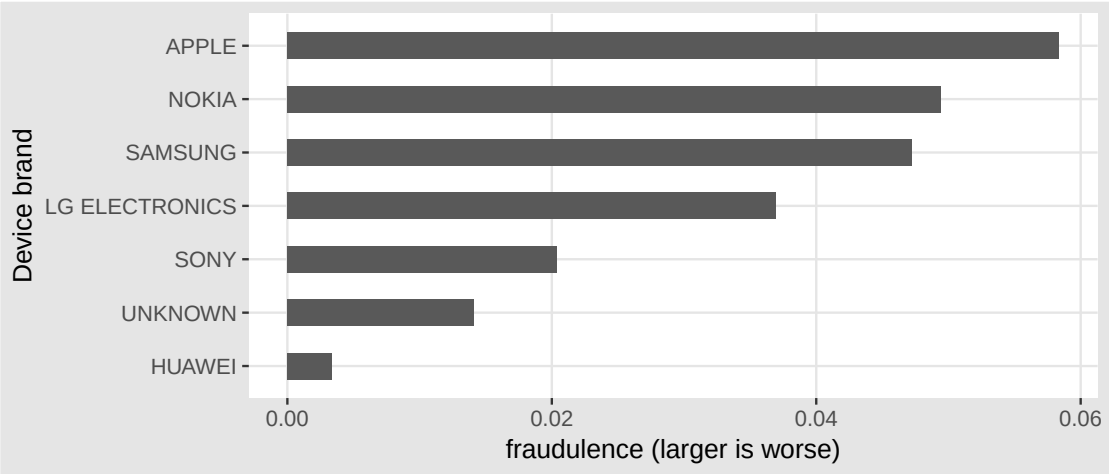


Figure 3.19: Fraudulence per device brand

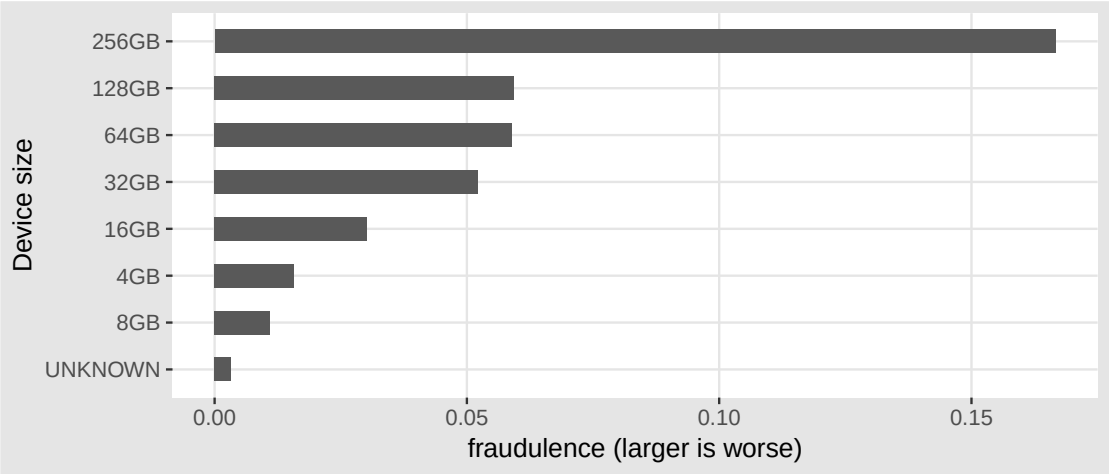


Figure 3.20: Larger storage on device induces a higher fraudulence

that neverpayers are mostly interested in the latest top smartphones. The quality of this field is not high, but still usable.

An initial up-front downpayment differentiates regular customers from neverpayers. See Figure 3.22. Our partner conducted experiments where a downpayment of 100 € was required. As the data shows, for regular customers, it is not the norm, but at least acceptable to pay up to nearly 300 € up front. However, it is clear that the expectation is to pay only a small amount of money up front.

3.3.5 Features about the contract

In the last section, we could see that current top smartphones are higher risk assets often included in a contract with a low down payment. When grouping contracts on

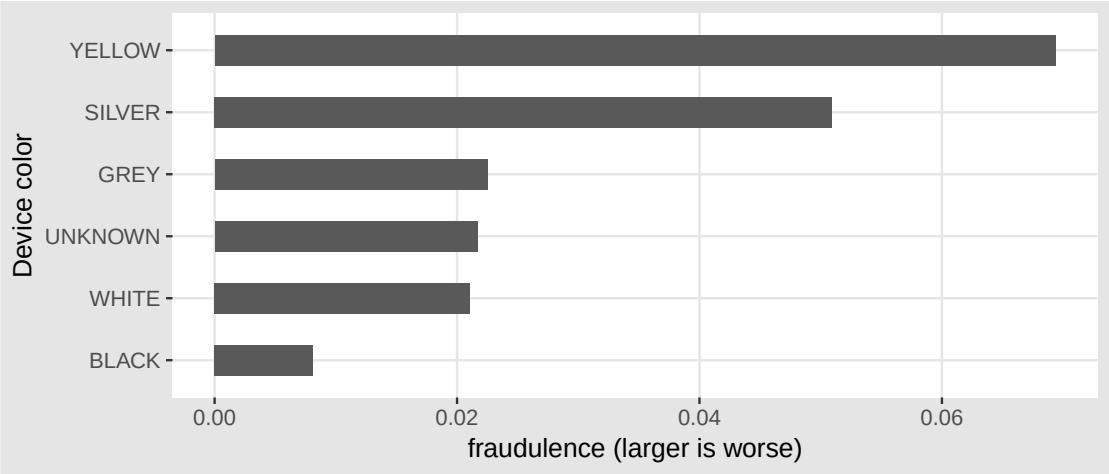


Figure 3.21: Yellow devices are more popular for neverpayers

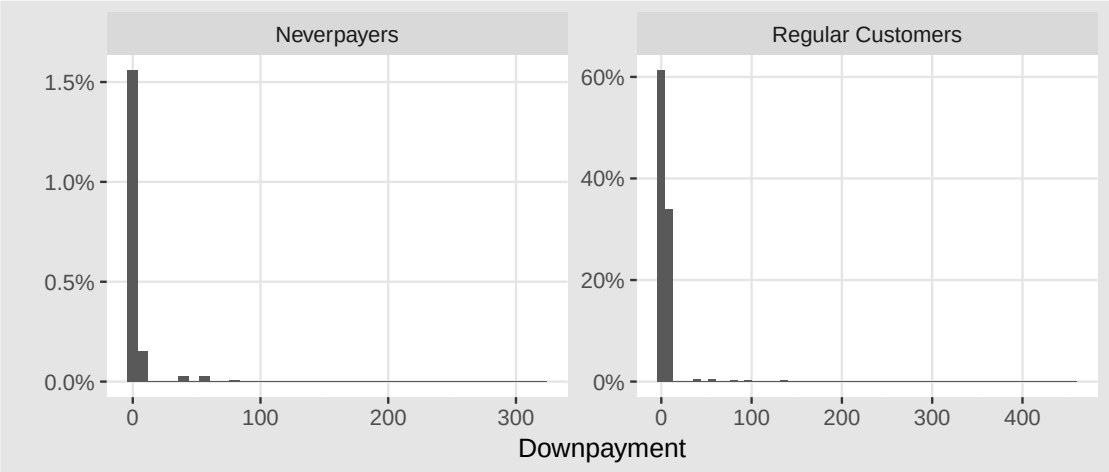


Figure 3.22: Fraudulence per down payment in local currency (€). Regular customers are inclined to pay more, even though no up front payment is the norm even for neverpayers.

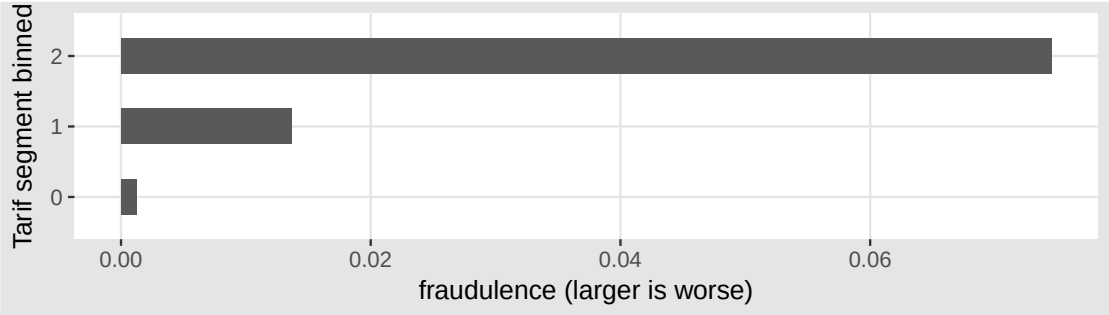


Figure 3.23: *0: low* $0 \leq t_i \leq 20$*, 1: medium* $20 < t_i \leq 40$ *2: high cost* $40 < t_i$*, where* t_i *denotes the tarif value bin of the i-th observation* of monthly fee for the contract per neverpayer. Pricier contracts allow to finance more expensive devices and therefore pose more risk.

Figure 3.24: Neverpayers per initial payment category. Payment forms pose a higher risk.

their monthly fee, Figure 3.23 depicts that the most expensive contracts show more risk for our partner. Only these expensive contracts include the latest top smartphones for a low or not existent initial fee.

When signing up, custosmers declare a means of first payment which should be used for paying the first bill. Depending on when the customer applies for a new contract, the bill might be due after round half to 2/3-rds of a month. During this time the means of payment can be changed. Our data only reflects the initial payment intention as visualized in Figure 3.24. As usually the first monthly fee is paid after the first month of usage the real payment category might have changed. One can see that payment forms pose a higher risk. This type of payment constitutes a less automated form of payment, i.e., the customer manually needs to initiate the transaction. Interestingly, only minimal risk applies for credit cards - so low that they do not even appear in the plot. Data quality for contract related features is rather good.

3.4 Data preparation

Cleaning the data according to You et al. (2016) who surveyed 186 data analysts consumes most of the time. Still, its importance is often overlooked. Garbage In Garbage Out (GIGO) is a widely accepted axiom in computer science. It implies that even the most sophisticated algorithm will perform poorly if the input is ill-prepared or of bad quality.

Several steps need to be performed to obtain useful data for later analysis. Jonge et al. (2013) name: localization of errors, correction, imputation of missing values. Data cleaning and feature engineering need to work together to provide an optimal data set for machine learning.

According to Underwood (2016) especially the feature engineering part from the preprocessing pipeline is more an art than a science.

3.4.1 Reading the data

The data originates from some classical database systems. Not all columns are neatly structured. Some free text fields exist, i.e., the return value from the credit scoring workflow which contains the score and some additional information how the score was generated. Unfortunately, special characters are contained in this response like newline characters. To be able to properly parse the data we resorted to using additional multi-character delimiters and special quotes and strip newline and other offending characters:

```
<feature_1>$$><$$<feature_2>
```

3.4.2 Missing and unseen value imputation

The cleaning pipeline needs not only to handle missing values but also identify unseen values for categorical data. Imputation of missing values depends on the feature. In general, a value which is not distorting the distribution is inserted. For categorical values, a new unknown label is introduced. In both cases, artificial columns are introduced which preserve the information that this value was previously missing.

As this cleaning process is meant to not only serve for analytical purposes but contribute to business decisions in real time we explicitly need to handle previously unseen values for categorical data. We choose to perform most frequent imputation here.

3.4.3 Removing corrupted observations

Unfortunately, the overall quality of the batch training data was not great as the query evolved organically. Some join must have introduced duplicate records. We needed to drop duplicates and decided to keep the last most current values here.

3.4.4 Categorical to numeric transformation

Many our attributes initially are not numeric. I.e., name, address, dealer and much more are categorical attributes. In general, machine learning algorithms work only with numeric data, so somehow we need to transform this data. Soukhavong (2017) concludes that categorical features with

- large cardinalities (over 1000) should be encoded binary
- small cardinalities (less than 1000) should be encoded numerically

Soukhavong (2017) outlines that the order imposed by standard numeric label encoding, i.e., encoding a sequence of `[cat, dog, cow]` to `[1, 2, 3]` which means imposing an implicit ordering, has no effect on tree-based learners as these do not calculate a distance between these values but rather only calculate a split point.

In general dummy/one-hot-encoding is a widespread scheme to deal with categorical data. However, one-hot-encoding only is suitable for categorical values with not too many levels as otherwise a too large matrix is created. Unfortunately, several implementations of current machine learning algorithms cannot use or have problems with sparse matrices which can result in a memory problem if too many columns are created. (UCLA IDRE Institute for Digital Research and Education, 2014) provide an R package and excellent documentation about other possible coding variants.

From our experience, domain-specific coding can be very helpful. Initially, we calculated the fraudulence per each group per each categorical column. The results can be seen in the plots of Section 3.3. As the initial strategy was to have high precision, adding measured fraudulence from the data which introduced bias turned out to achieve this goal quite well. Later on, the strategy of our partner changed to identify at least 50% of neverpayers while still trying to minimize false positives, i.e., actually good customers falsely identified as neverpayers. Therefore we switched to numeric encoding for the final strategy.

3.4.5 Feature engineering

Reminder: we can only use features available at T_0 at the POS before the device is handed out and potentially lost.

Anomalies in distributions

Some of the features, e.g., *nationality*, contain mostly values of a single category. In the case of the example: *Austrian*. We try to help the model to easier handle this by introducing binary columns.

In case of numeric anomalies, we resort to using *log1p*[1] to prevent an anomalously large value to distort the numeric distances.

Distance to border

Our dataset contains multiple addresses. One of these belongs to the customer. We use reverse geocoding to get coordinates for this customer. Additionally, a shape-file of Austria geographically references its borders. We then calculate the distance between each customer and the border.

Dates & time

Dates and timestamps include a lot of hidden but valuable features which manually need to be engineered. A timestamp `2017-01-01 01:01:01` may include periodicities to a fine granular level of seconds or more. We extract

- month
- day
- day of week
- quarter
- hour
- minute

as separate features.

Additionally, birthdates are converted to an age variable.

Holidays and public holidays have a significant impact on sales. We calculate the distance of a sales transaction (backward and forwards) to Austrian and foreign holidays and public holidays as multiple separate features.

Binning

Some variables are binned to reduce the variance the machine learning procedure has to learn. For example the title is reduce to `male-withTitle, female-withTitle, male-noTitle, female-noTitle`. For numeric values, e.g., the result of the credit scoring procedure, additional bins are introduced.

[1] *log1p* adds 1 to prevent problems with the logarithm around 0 and therefore is more practical in real-world situations than an ordinary logarithm.

Distances between names

The dataset contains several names:

- customer
- bank account
- e-mail
- e-mail electronic billing
- seller

A variety of string distance measurements is calculated here trying to capture similarity.

Interactions

Interactions between two or more features are extracted as separate features. There are different timestamps in the data. We calculate the difference between these, for example as mentioned before, for the holidays or between different timestamps associated with the transaction.

Initially, we were experimenting with polynomial interactions, but that resulted in an overly time-consuming training process of the model without gaining sufficient improvement of the results.

Additional data

Open data sources are used to add additional information, especially to categorical features.

3.5 Dealing with highly skewed data

The dataset used in this analysis is highly skewed. The positive minority class of never paying customers is of very low cardinality and constitutes only about 1.9% of all samples.

Typically, a model is tested via cross-validation. To make sure that observations from both classes are available in each fold, a stratified cross-validation procedure can be used. As shown in Figure 3.25, we do not want to randomly pick observations when splitting the data, but rather focus on the business problem at hand and predict the fourth month ahead as the three months in between do not constitute legally binding ground truth data. Therefore we use a custom time series cross-validation splitting strategy with the desired properties which resemble reality closer as at some point the model should be deployed into a real-time production environment. We can not apply stratification as the splits are performed per time windows.

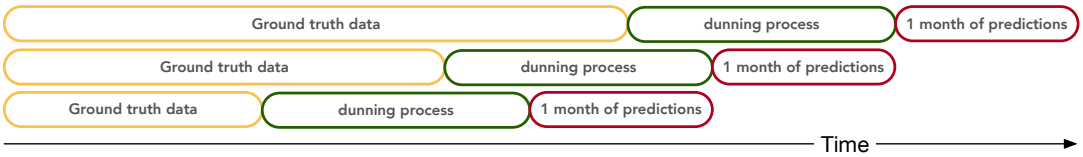


Figure 3.25: Custom time series cross-validation which predicts the fourth month ahead as three months in between are legally not yet considered ground truth data.

3.5.1 Danger of skewed data

In general, machine learning models assume an even class distribution for the metrics which are optimized. Usually, a metric like accuracy is employed to evaluate the model. As Elitedatascience (2017) demonstrates, this can lead to a simple model which can get impressively high scores, i.e., only predicting classes of label 0 which resemble regular customers. However, it is evident that this type of model cannot be used and will not meet any business goals.

3.5.2 Resampling

Without changing the evaluation criterion, the same model can be used as long as the distribution of the data is altered. Therefore, it is widely used in industry (Padmaja et al., 2007; Hollmén, 2000; Mählmann, 2010). It has been shown, that using a $50:50$ resampled distribution for training significantly reduces the amount of loss (Chan et al., 1998).

Rebalancing can be achieved through up-, downsampling or hybrid approaches like Synthetic Minority Over-sampling Technique (SMOTE) Chawla et al. (2002). The benefit is that for rebalanced data, off-the-shelf machine learning classifiers can be utilized as the class distributions will be roughly equal (Kuhn et al., 2013).

Up-sampling minority class

Up-sampling means generating more observations of the minority class by applying replication of scarce observations.

However, adding new observations by duplication can distort the decision boundary and result in overfitting (Vida, 2016).

There also exist smarter sampling strategies for upsampling of the minority class. As visualized in Figure 3.26, upsampling of the minority class by replication is not as effective as applying synthetic generation according to the glssmote algorithm. Synthetic observations of the minority class are filling the space of a convex hull with new observations.

A common mistake is to perform upsampling on the whole dataset. This cannot lead to any sensible classification results as the test/validation datasets are resampled as well (Altini, 2015).

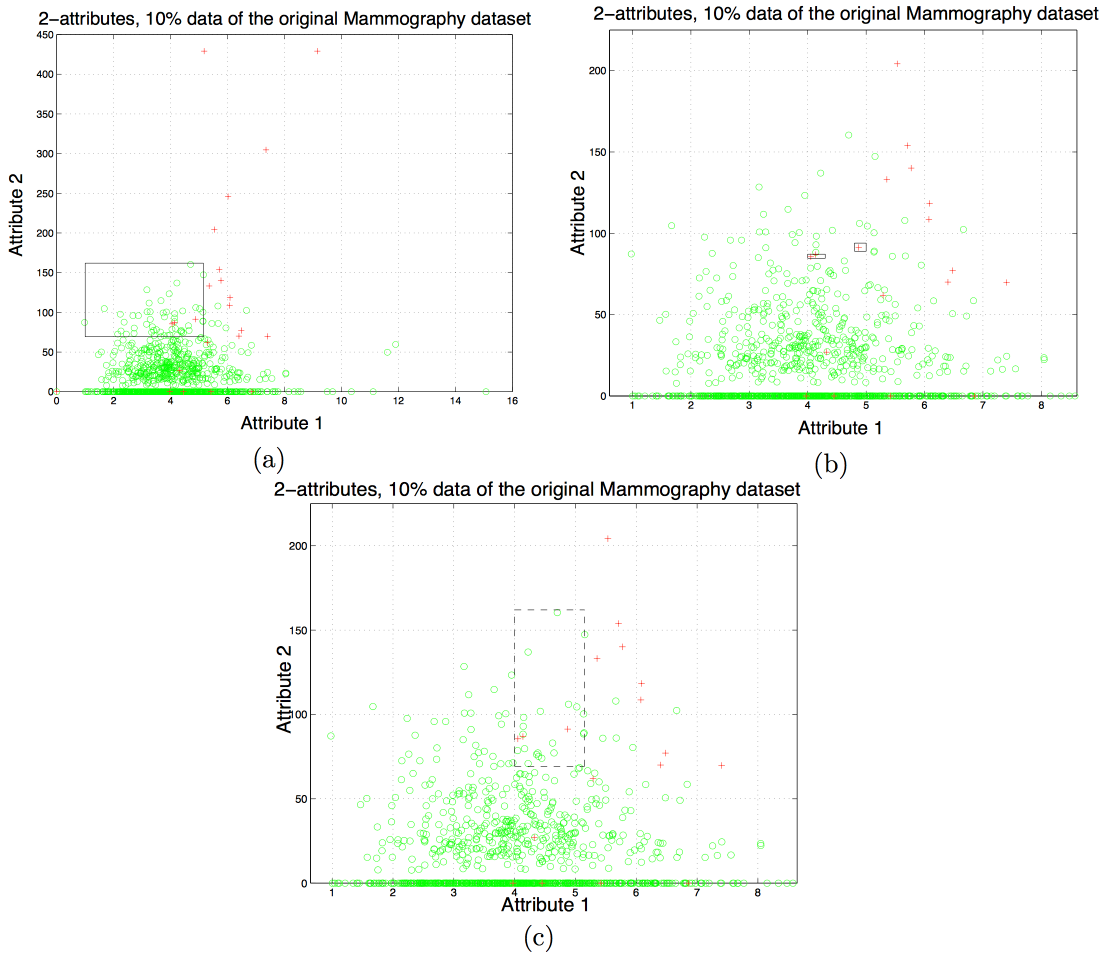


Figure 3.26: a) Decision region where minority class denoted by + reside after building a decision tree marked by a rectangle. b) A zoomed-in view of the selected minority class samples. Small rectangles show the decision regions after oversampling the minority class with replication. c) A zoomed-in view of the selected minority class samples where the decision region after oversampling the minority class with synthetic generation is depicted. Chawla et al. (2002, Figure 3)

.

Down-sampling majority class

A second simple possibility is to downsample the majority class. Several strategies exist to choose the samples, i.e., random, furthest away from the decision boundary. According to Japkowicz (2000) simple random sampling works just as good as any more elaborate methodology. The more the data set is imbalanced, the more possibly useful information is discarded (Altini, 2015). This methodology is only advisable for massive datasets.

Combined approaches

Up- and down-sampling strategies can be combined. (Batista et al., 2003) note that SMOTE can be accompanied by intelligent downsampling of the majority class based on edit nearest neighbors or Tomek links where certain neighborhoods of a k-nearest neighbor clustering are applied.

We decided not to use such data rebalancing methodologies, because:

- results did not improve when using SMOTE
- no good rebalancing implementation exists in python which allows for fast training of classifiers or handling of sparse matrices at the point of writing this thesis
- cost-based methods should allow us to report more meaningful metrics and focus better on the individual value at risk

3.5.3 Skew invariant performance metric

Our problem is a classification problem. Labels of the binary class are predicted, and a confusion matrix can be created from the predictions. The different types of errors, i.e., false positives and false negatives are visualized in Figure 3.27. Several metrics which combine the four values from the confusion matrix into a single value can be calculated, precision, recall or accuracy are fairly common.

For an evaluation of a general classification learning problem, these work great, but in case of severe class imbalances or possibly uneven cost distribution for the classes, these are no longer well-suited (Correa Bahnsen, Aouada, et al., 2016; Ganganwar, 2012; Jeni et al., 2013; Monard et al., 2002; Provost et al., 1997; Kuhn et al., 2013).

Firstly, we want to stress the importance of choosing the right metric. In our case, the department of customer finance was piloting their first machine learning project. Only limited knowledge about the so far unknown domain of artificial intelligence and machine learning was available and some evaluation criterion needed to be chosen. At first, accuracy was used to evaluate the model. This is standard practice and often is not questioned enough. However, it is bad as

- business needs are not reflected by this metric,

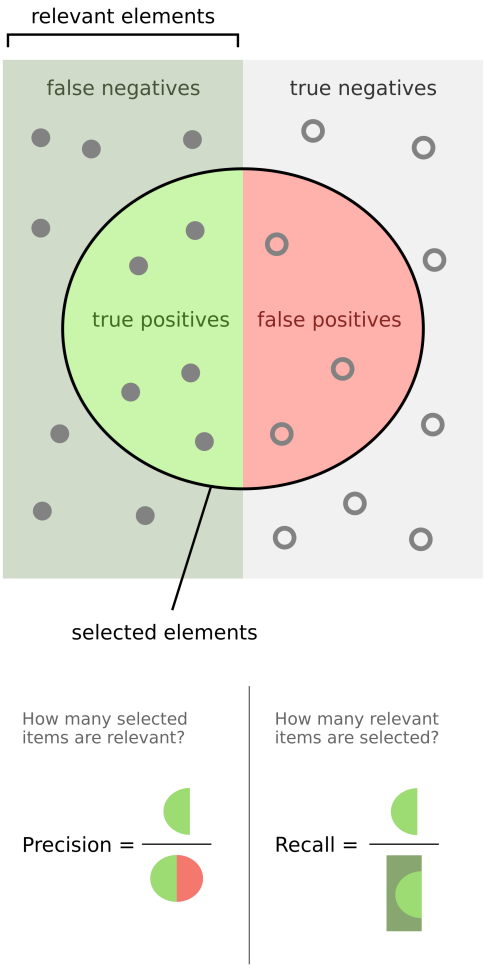


Figure 3.27: Error types from confusion matrix (Walber, 2017)

- data skew dramatically distorts the results of this metric.

When computing a metric per class it gets more interesting as the average about all (in our case 0,1) classes hide important details which gets even more important for multi-class classification problems, but already shows its impacts when the class distribution is highly imbalanced as it is the case here.

Metrics which handle data skew gracefully is, for example, *AUC (area under the curve)* (Elitedatascience, 2017).

From experimenting with our dataset, we conclude that Cohen's kappa and F_β score, ($\beta > 1$) is a good fit as well, as the imbalanced nature of the data set is not resulting in distorted scoring results reported by the metric. Especially the F_2 score which constitutes the weighted harmonic mean between false negatives and false positives for more weight

on finding relevant observations is defined as:

$$F_\beta = \frac{(1+\beta^2) * truePositive)}{(1+\beta^2)truePositive + \beta^2 * falseNegative + falsePositive}$$

where $\beta = 2$.

Kappa is a single scalar metric, where a value of 1 is the best and -1 the worst value. Kappa borrows from information theory's concept of the expected information (Rbx, 2014; Said Bleik, 2016; „Kappa“ 2007). Random chance is considered.

Multiple variants exist Cohen's, Fleiss's and quadratic (weighted) Kappa, where the last one is best suited for multi-class classification where a higher penalty will be applied for a class label further away from the real class.

Often off the shelf cost, insensitive algorithms are used to perform classification tasks (Bahnsen, Aouada, et al., 2014). For many binary classifiers, the real probability of the event is overlooked as they only focus on the separation between positive and negative examples, but not on the real cost which affects business (Cohen et al., 2004).

Recently interest in better-tailored evaluation metrics has risen. (Correa Bahnsen, Aouada, et al., 2016) propose a cost-based evaluation tailored to the underlying business problem. Later, in Section 3.5.6 we explore these concepts in more detail and adapt them to the telecommunications industry in Section 3.8.

A good read is[2] which explains several metrics in detail.

3.5.4 Choice of classifier

Tree-based classifiers possibly handle imbalanced datasets gracefully (Elitedatascience, 2017). Consider experimenting with such a classifier (decision tree, random forest, gradient boosting) not only for easier handling of skewed data but also for excellent handling of categorical data.

3.5.5 Reformulation as anomaly detection

Labelling fraud is hard, but recognizing abnormal patterns is a great task for a computer. The classification problem can be reformulated in an anomaly detection problem. This can be especially helpful to detect still unknown patterns of fraud which also deviate from a regular user.

Andrew Ng, an Associate Professor at Stanford, explains the difference between anomaly detection and classification problems[3]. He denotes that:

- imbalanced classes are common for anomaly detection

[2] http://www.r-bloggers.com/a-budget-of-classifier-evaluation-measures/

[3] https://www.coursera.org/learn/machine-learning/lecture/Rkc5x/anomaly-detection-vs-supervised-learning

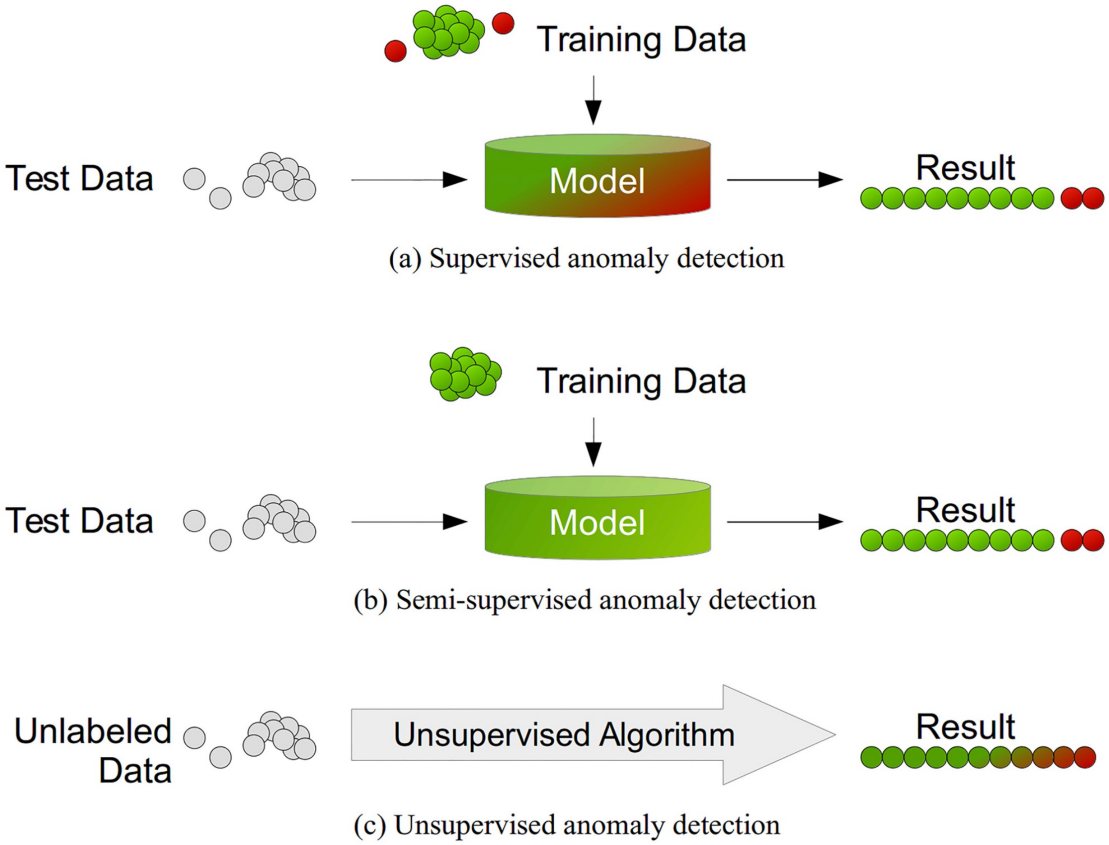


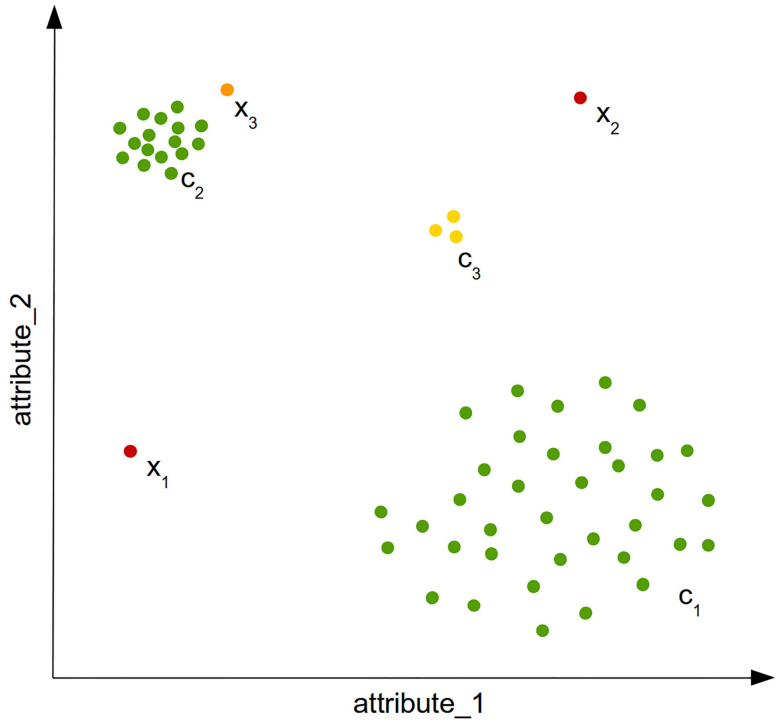
Figure 3.28: Anomaly detection setups (Goldstein et al., 2016, Figure 1).

- different types of anomalies exist. Not all will be covered in the training sample, but still, it is desirable to catch and detect yet unknown anomalies

Goldstein et al. (2016) explain different anomaly detection setups as shown in Figure 3.28

- *Supervised anomaly detection a)* is very similar to classical classification, and fully labelled train and test datasets are used. Due to highly imbalanced classes only some algorithms can deal with this problem or special modifications are required. However, practically it is not very relevant due to the assumption of known anomalies which are labelled correctly.
- *Semi-supervised anomaly detection b)* uses a train dataset of only normal data, whereas only the test dataset contains the anomalies. Any deviations from normality should be recognized by the algorithm
- *Unsupervised anomaly detection c)* tries to infer structure in the data itself useful for scoring.

Figure 3.29: Types of anomalies Goldstein et al. (2016, Figure 2).



According to Goldstein et al. (2016) the output of an anomaly detection algorithm is either labels or scores. Labels are used for supervised anomaly detection, whereas scores which give a little bit more detail are more common in semi- or unsupervised outlier detection tasks.

Chandola et al. (2009) categorize anomalies into the classes described below. These are visualized in Figure 3.29:

- *point:* an individual instance is anomalous: x_1 and x_2 can be identified as outliers easily. Both are far away from any dense area.
- *context:* a data instance is anomalous within a context: A monthly time series with spikes in certain months is a good example. In this figure, x_3 represents such an anomaly: from a global context, it might be considered normal, relative to c_2. However, when looking only at c_2 it becomes apparent that x_3 is an outlier.
- *collective:* describes a collection of related data points as anomalous. In the Figure, c_3 as a micro-cluster fits that description as individual instances of a collective anomaly is not an outlier by itself. This last type is more suited to sequential, graph or spatial data where, e.g., for an EEG time series the heartbeat will restart after the revival of the patient.

Several challenges involved with anomaly detection are described by Chandola et al. (2009). The following subset will be of particular importance four our anomaly detection problem:

- anomalies and what is considered normal evolve over time
- malicious adversaries adopt and try to look normal
- distinguishing noise from real anomalies is hard

See Section 3.6.3 on page 49 for technical details about the algorithms chosen in our solution. In the next section, we will introduce several evaluation metrics suited for classification problems.

3.5.6 Cost based penalties

Tailoring the loss function to the problem domain with cost can be helpful to overcome data skew.

Class based cost

Most classifiers offer an option to balance class weights automatically. Here, the majority class is reweighed according to the amount of proportional overrepresentation.

Experimentation with cost per class might help to tailor the loss to the problem (Elitedatascience, 2017). Other values than the one chosen automatically might be helpful.

When tuned correctly, this can significantly help to overcome data skew, but some desirable properties are still missing. Example dependent cost based classification will support these.

Example dependent cost

As already outlined in Section 3.5.3, a good evaluation metric should be easily explainable to domain experts and be invariant to data skew. Example dependent cost matrices could fulfill these promises and offer a better understanding of the situation as the individual risk can be evaluated better as the realities of business, e.g., market invest cost and the expected profits of a particular client are taken into account. Similar to credit card fraud detection, the cost of false positives for fraudulent activities in telecommunications is different than the cost for false negatives (Correa Bahnsen, Aouada, et al., 2016). A falsely flagged customer will cause administrative overhead and may spread bad publicity of the company for denying him access (Shawe-Taylor, 1999). However, as good customers only make money over time, i.e., when they stay long in the contract, additional money is lost as opportunity cost. Not being able to detect fraud means losing money (Hilas et al., 2005). Additionally, there is no constant cost difference between

false positives and false negatives, as the consumption of the provided services varies considerably (Estêvez et al., 2006).

Bahnsen pioneered the concept of example dependent cost based classification in a variety of papers (Van Vlasselaer et al., 2015; Bahnsen, Aouada, et al., 2015a; Bahnsen, Aouada, et al., 2015b; Correa Bahnsen, Stojanovic, et al., 2014; Correa Bahnsen, Aouada, et al., 2015; Bahnsen, Aouada, et al., 2014; Correa Bahnsen, Aouada, et al., 2016) for the detection of fraud in the credit card transactions.

We will take the proposed cost matrix and adapt them to the telecommunications industry. Then his cost-based algorithms are compared with other cost insensitive machine learning models.

3.6 Models

A variety of models is compared on this dataset. The following section describes the mathematic intentions behind each model. Only the models used later on when comparing results are described below.

Indeed, it would be interesting to integrate clustering, Gaussian models, neural networks or single class approaches like one class support vector machines into a big ensemble model, but as the focus is to analyze cost-based methods we cannot include them. When deploying a model for real though, they might be an excellent addition. Therefore, we assume a supervised model, where $\mathbf{y}$ denotes the vector of n binary labels for each observation, where 1 constitutes the positive class labels, i.e., neverpayers we want to detect and 0 denotes a regular customer. $\mathbf{X}$ is the matrix of n observations with p features. y_i is the i-th observation of $\boldsymbol{y}$ and $\boldsymbol{x}_i$ the i-th observation of the p-dimensional space $\boldsymbol{X}$.

3.6.1 Logistic regression

Logistic regression is a particular case of generalized linear models. Commonly it is used for classification problems. In particular, it is famous for two-class classification. Researchers like this method due to its simplicity and the possibility to conduct formal statistical tests to decide if a variable is relevant (Whitrow et al., 2009). Similar to the standard *least squares* estimator, the sum of the deviances is minimized, which is based on the maximum likelihood approach (Kuhn et al., 2013).

The *odds ratio* can be defined as $\frac{p}{1-p}$ (Raschka, 2015), where p represents the probability of the event. In our case, this means a neverpayer. The *logit* function can be defined as:

$$logit(p) = \log\left(\frac{p}{1-p}\right) = \log\left(\frac{P(y=1|\mathbf{x})}{P(y=0|\mathbf{x})}\right) = \beta_0 + \boldsymbol{\beta}_1^{\mathsf{T}}\mathbf{x} = z \tag{3.1}$$

It is just the logarithm of the odds ratio. As p is a probability, the input of the logit function is in the range $[0, 1]$ which is transformed to the entire range of $\mathbb{R}$. This eases the

formulation of a simple linear regression model which denotes the conditional probability that a sample belongs to the class of never payers. The estimated probability of a sample belonging to class $y = 1$ is given as

$$\hat{p} = \frac{1}{1 - e^{-\hat{z}}} \tag{3.2}$$

with the estimated values

$$\hat{z} = \hat{\beta}_0 + \hat{\boldsymbol{\beta}}_1^{\mathsf{T}} \mathbf{x} \tag{3.3}$$

The output can be interpreted as a binary classification to predict class labels like:

$$\hat{y} = \begin{cases} 1 & \text{if } \hat{p} \geq \text{cutoff} \\ 0 & \text{otherwise} \end{cases}$$

where usually cutoff = 0.5. This value can be tuned to bias scoring into a direction and suit business needs. In this thesis we use the default value of 0.5 as less bias into a direction is introduced.

3.6.2 Rule based approaches

Without machine learning classifiers the customer finance department of our partner already manually developed rules. Rule-based approaches efficiently capture non-linear patterns in the data. Generating those rules automatically can be seen as the next step.

Criminisi (2011) provides a good overview of different types of trees available.

Decision tree

Decision trees automatically identify classification rules from the data by partitioning the feature space into a space of rectangles and fit a simple model for each one (Hastie et al., 2001).

For a node m in one of the rectangles, i.e., region m, R_m, the number of observations in this region is n_m.

$$\hat{p}_{mk} = \frac{1}{n_m} \sum_{x_i \in R_m} I(y_i = k)$$

denotes the fraction of the observations in R_m which are of class label k. The node is assigned to class $k(m) = \text{argmax}_k \hat{p}_{mk}$, i.e., the most frequent one.

Figure 3.30 outlines the inner workings of a decision tree by showing how a family member is classified as someone who is a gamer. From a root, decision rules are inferred step-by-step until a leaf node is reached which only contains observations of a single class. Note, the Figure depicts a classification and regression tree (CART) and not a regular decision tree as the final leaves do not already constitute the decision, but rather real valued numerical values which allow for finer grained decisions and optimizations in the implementation. Similar to a manual score card, binary split points are created

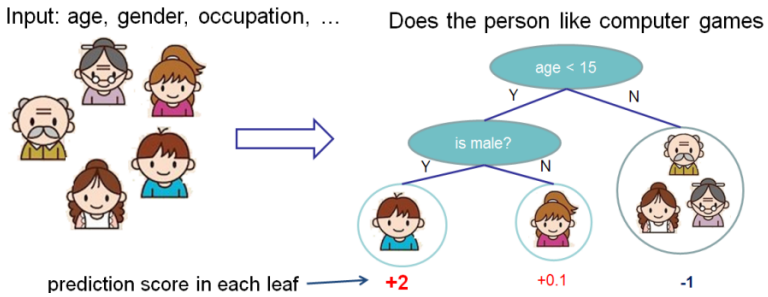


Figure 3.30: Explanation of a decision tree by classifying a family into gamers and non gamers (Chen et al., 2016).

recursively to partition the data into smaller groups. The split is performed according to a criterion which defines the optimality of the splits.

Commonly used criteria to calculate an optimal split are (Criminisi, 2011):

- misclassification rate $\frac{1}{n_m} \sum_{i \in R_m} I(y_i \neq k(m)) = 1 - \hat{p}_{mk(m)}$
- gini index $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^{\text{classes}} \hat{p}_{mk}(1 - \hat{p}_{mk})$
- cross entropy or deviance $\sum_{k=1}^{\text{classes}} \hat{p}_{mk} \log \hat{p}_{mk}$

Here, classes is the number of classes. In our case of binary classification where classes = 2 the three criteria are: $1 - max(p, 1-p)$, $2p(1-p)$ and $-p \log p - (1-p) \log(1-p)$

Decision trees are rather easy to understand from a businesses perspective and gracefully handle categorical or missing data as well as non-normalized features or imbalanced datasets. Additionally, they do not require the tuning of many hyper-parameters.

The downside is that a single decision tree may turn out to be unstable when fitted on new data especially when the decision boundaries were overfitted. This results in a high variance of classification boundaries and is not desirable behavior. Pruning can be used to prevent overfitting (Hastie et al., 2001).

Random forest

To overcome these issues, multiple trees can be fitted as introduced by Breiman (2001). Better generalization is achieved by de-correlating predictions, i.e., randomizing the data and fitting a tree each time, which later on are all averaged for an ensemble prediction.

Figure 3.31 outlines an example which shows that different trees do not necessarily use the same features. In fact, the result will be more stable when column- and row-wise subsampling (bagging) is applied. The scores obtained when predicting new values are summed up for each observation in case of CART and afterwards a class label can be determined or majority voting is used. This results in an easy to use and great classifier

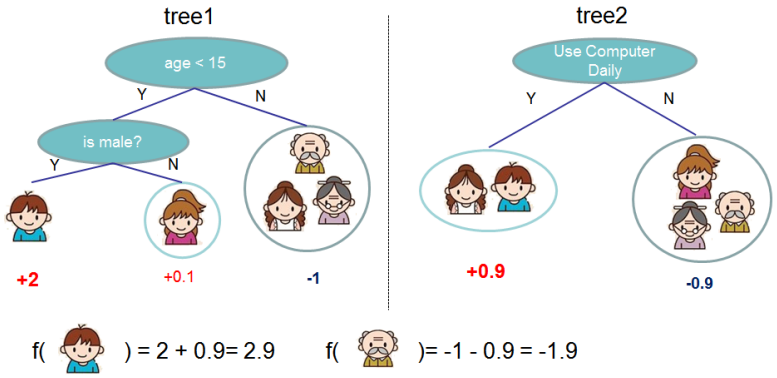


Figure 3.31: Two trees contributing to the overall scoring of an ensemble tree based model (Chen et al., 2016).

so that some claim that random forests are their go to model for a wide range of machine learning problems (Deeb, 2017).

Gradient boosted trees

Instead of fitting the trees at random they are fitted in a way which minimizes errors of previous trees. Figure 3.31 already outlines how trees can complement each other. An initially weak learner (stump of decision tree) can grow into a powerful ensemble when iteratively fitting models of the residuals (boosting) to compensate the errors.

According to (Chen et al., 2016) the objective to be minimized can be written as:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\boldsymbol{x}_i), f_k \in \mathcal{F}$$

where K denotes the number of trees, f_k is a function in the functional space $\mathcal{F}$ which itself resembles the set of all possible CART trees. The objective to be optimized can be written as $\sum_{i\geq 1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$ which includes the training loss $l(y_i, \hat{y}_i)$ and regularization function $\Omega(f_k)$. One can see that from a perspective of the model random forests and gradient boosted trees are the same. The sole difference is the strategy applied when training the trees.

Prediction $\hat{y_i}^{(0)}$ at step t is defined as:

$$\begin{aligned}
\hat{y}_i^{(t)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(\boldsymbol{x}_i) = \hat{y}_i^{(0)} + f_1(\boldsymbol{x}_i) \\
\hat{y}_i^{(2)} &= f_1(\boldsymbol{x}_i) + f_2(\boldsymbol{x}_i) = \hat{y}_i^{(1)} + f_2(\boldsymbol{x}_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(\boldsymbol{x}_i) = \hat{y}_i^{(t-1)} + f_t(\boldsymbol{x}_i)
\end{aligned}$$

and the tree at each step which minimizes the objective is defined as

$$\begin{aligned}\text{obj}^{(t)} &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i) \\ &= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\boldsymbol{x}_i)) + \Omega(f_t) + constant\end{aligned}$$

Industrial grade implementations are for example scikit-learn's *GradientBoostingClassifier*, xgboost[4], LightGBM[5] and recently catboost[6]. *LightGBM* was used for this thesis as *catboost* was introduced only very recently and still as some bugs to fix.

These implementations are improved continuously. Recently, histogram-based approximative algorithms, parallel learning, improvements on how trees are grown or GPU support were added to improve models. In the case of *LightGBM*, [7] lists their respective improvements to the tree building process.

3.6.3 Example dependent cost-based approaches

The cost, i.e., value at risk for each single observation is taken into account when using an example dependent classifier. Simpler variants take the output from an existing model and then only compute cost-based metrics from it. More complex models directly try to optimize the cost or respectively the savings. Bahnsen pioneered this types of models in (Bahnsen, Aouada, et al., 2015a; Bahnsen, Aouada, et al., 2015b; Correa Bahnsen, Stojanovic, et al., 2014; Correa Bahnsen, Aouada, et al., 2015; Bahnsen, Aouada, et al., 2014; Correa Bahnsen, Aouada, et al., 2016).

Evaluation of example dependent cost-based models - savings score

Fraud often has a unique cost associated with the different classification error. In fact, one usually cannot generalize the prediction error as an average per class but should individually assess the risk (Bahnsen, Stojanovic, et al., 2013). They propose an example dependent cost matrix to adequately consider the value at risk for each observation as the amount at risk for credit card transactions differs a lot.

The amount of cost can easily be considered a metric for the model defined by

$$\text{Cost}(f(S)) = \sum_{i=1}^{N} (y_i(c_i C_{TP_i} + (1 - c_i)C_{FN_i}) + (1 - y_i)(c_i C_{FP_i} + (1 - c_i)C_{TN_i}))$$

as defined in Bahnsen, Aouada, et al. (2014). $C_{TP_i} = C_{TN_i} = 0$ which denotes the cost for the correct predictions and is assumed to have no additional cost for each observation

[4] `https://github.com/dmlc/xgboost` (Chen et al., 2016)
[5] `https://github.com/Microsoft/LightGBM`
[6] `https://github.com/catboost/catboost`
[7] `https://github.com/Microsoft/LightGBM/wiki/Features`

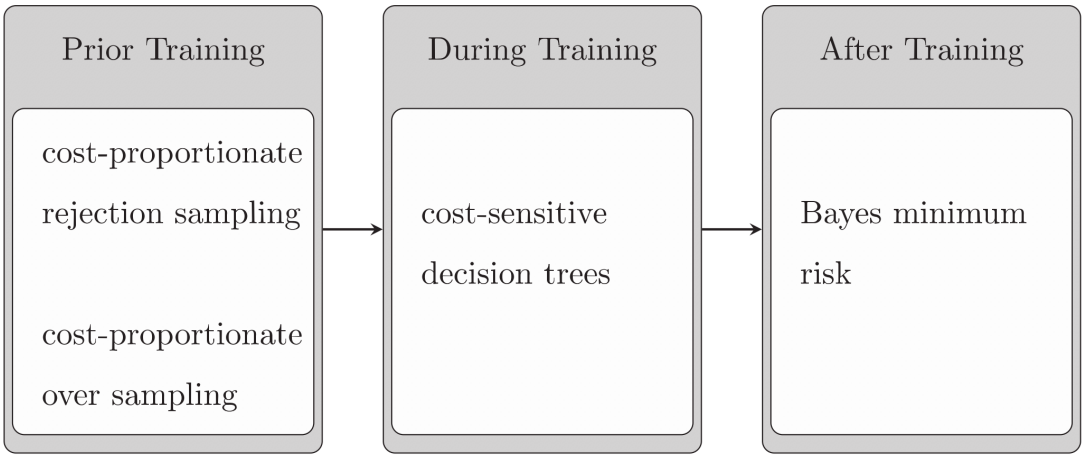


Figure 3.32: Example-dependent cost-sensitive algorithms grouped by the stage where they are used in a classification task (Bahnsen, Aouada, et al., 2015b, Fig 1).

i associated with it. C_{FP_i} is the cost for a false positive, C_{FN_i} for a false negative of observation i and S is a set of N samples $i, N = |S|$. The associated cost for each observation can be considered a tuning parameter.

Assuming all observations are classified to the class with the lowest cost $\text{Cost}_l(S) = \min\{\text{Cost}(f_0(S)), \text{Cost}(f_1(S))\}$, f_0 denotes the classifier which classifies all the observations in S to class label 0, f_1 to 1. The cost improvement compared to using no classifier at all can be expressed as the cost savings compared with $\text{Cost}_l(S)$:

$$\text{Savings}(f(S)) = \frac{\text{Cost}(f(S)) - \text{Cost}_l(S)}{\text{Cost}_l(S)} \tag{3.4}$$

The cost matrix proposed by us for the telecommunication industry is defined in section 3.8.

The methods introduced below not only use cost or savings for evaluating the classification result but also during training. Figure 3.32 outlines the stages where the different algorithms can be used for a classification task. Reweighing the training observations based on their costs is the most common approach performed either via cost-proportionate rejection-sampling (Zadrozny et al., 2003) or oversampling (Elkan, 2001).

Both do not use the full cost matrix but only consider misclassification cost. Also, cost-proportionate over-sampling increases training time and may overfit the model (Drummond et al., 2003).

Bayes Minimum Risk (BMR)

BMR is using the predictions of class probabilities of an existing classifier and tries to maximize the expected value of savings calculated from an example-dependent cost matrix.

Ghosh et al. (2006) define the BMR classifier as a decision model which quantifies trade-offs of different possible decisions using probabilities and the respectively associated cost of the decision. In our case, the decision is binary: accept or decline the new customer, i.e. classify as fraudulent p_f or legitimate p_l. The risk with a customer is predicted as fraudulent can be calculated as

$$R(p_f|\boldsymbol{x_i}) = l(p_f|y_f)P(p_f|\boldsymbol{x_i}) + l(p_f|y_l)P(p_l|\boldsymbol{x_i})$$

when predicted as legitimate as

$$R(p_l|\boldsymbol{x_i}) = l(P_l|y_l)P(p_l|\boldsymbol{x_i}) + l(p_l|y_f)P(p_f|\boldsymbol{x_i}),$$

where y_f and y_l are the true labels for fraudulent and legitimate customers, respectively, as defined by (Bahnsen, Stojanovic, et al., 2013).

The estimated probability $P(p_l|\boldsymbol{x_i})$ denotes the probability of a new customer being legitimate, $P(p_f|\boldsymbol{x_i})$ the probability of fraudulence given the features of $\boldsymbol{x_i}$. $l(a, b)$ describes the loss function when a customer is predicted as a but really would have been b. Given both risk values, a new customer application is classified as fraudulent if the risk of classifying as fraud is lower than classified as a legitimate user, $R(p_f|\boldsymbol{x_i}) < R(p_l|\boldsymbol{x_i})$, which in our case is equal to the cost defined by the cost matrix.

A probability should not only separate well between negative and positive samples but also assess the real probability of the event (Chakrabarti, 2004). The next algorithms truly take this into account.

Example-depdendent cost-sensitive logistic regression for credit scoring

The original definition of cost insensitive logistic regression is explained in Equation (3.1).

$\hat{p}$ as defined in Equation (3.2) depends on $\hat{z}$ which is defined in Equation (3.3) can also be written as $\hat{p}_{\hat{z}}$. This formula describes the logistic sigmoid function dependent on the parameter $\hat{z}$. The parameters which minimize the cost function are searched, which normally is defined as negative log likelihood

$$J(\hat{z}) = \frac{1}{N}\sum_{i=1}^{n} N J_i(\hat{z})$$

where

$$J_i(\hat{z}) = -y_i \log(\hat{p}_{\hat{z}}(\boldsymbol{x}_i)) - (1 - y_i)\log(1 - \hat{p}_{\hat{z}}(\boldsymbol{x}_i))$$

Bahnsen, Aouada, et al. (2014) show how cost sensitivity can be embedded here by replacing $J_i(\hat{z})$ with its respective cost which can be merged into a function dependent on new costs:

$$J_i^c(\hat{z}) = \frac{1}{N}\sum_{i=1}^{N} (y_i(\hat{p}_{\hat{z}}(\boldsymbol{x}_i)C_{TP_i} + (1 - \hat{p}_{\hat{z}}(\boldsymbol{x}_i))C_{FN_i} + (1 - y_i)(\hat{p}_{\hat{z}}(\boldsymbol{x}_i)C_{FP_i}) + (1 - \hat{p}_{\hat{z}}(\boldsymbol{x}_i))C_{TN_i})$$

where usually the cost for correct classification is assumed to be 0, $C_{TP_i} = C_{TN_i} \approx 0$, as no additional cost is caused by the model.

Cost-sensitive decision trees

By using measures such as entropy, misclassification or gini, the class distribution is influencing the result of the metric. All of these minimize misclassification rate, but as shown by Bahnsen, Stojanovic, et al. (2013) this leads to different results than minimizing cost. Bahnsen, Aouada, et al. (2015b) and Bahnsen, Aouada, et al. (2015a) propose a new split criterion based on example-dependent cost which does not optimize accuracy. They claim that their method is generating smaller decision trees compared to a cost-insensitive tree in about a fifth of the time.

The cost-based impurity measure is defined as the minimal cost of all the observations in a leaf when all observations in a leaf are classified both as negative using f_0 and positive using f_1 as defined by

$$I_c(S) = \min\{\text{Cost}(f_0(S)), \text{Cost}(f_1(S))\}$$

which evaluates to the lowest expected cost, where

$$f(S) = \begin{cases} 0 & \text{if Cost}(f_0(S)) \leq \text{Cost}(f_1(S)) \\ 1 & \text{otherwise} \end{cases}$$

and applied in the calculation of the split using the impurity of S minus the weighted impurity of each leaf by applying the splitting rule on feature $\boldsymbol{x^j}$ on value l^j:

$$\text{Gain}(\boldsymbol{x}^j, l^j) = I_c(S) - \frac{|S^l|}{|S|} I_c(S^l) - \frac{|S^r|}{|S|} I_c(S^r)$$

where $|\cdot|$ defines the cardinality and the two sets S^l and S^r are defined by:

$$S^l = \{\boldsymbol{x}_i^* | \boldsymbol{x}_i^* \in S \wedge x_i^j \leq l^j\}, S^r = \{\boldsymbol{x}_i^* | \boldsymbol{x}_i^* \in S \wedge x_i^j > l^j\}$$

$\boldsymbol{x}_i^j$ denotes the j-th feature of $\boldsymbol{x_i}$ and l^j is a value for which holds: $\min(\boldsymbol{x}^j) \leq l^j < \max(\boldsymbol{x}^j)$. The values of $(\boldsymbol{x}^j, l^j)$ is chosen to maximize the splitting criteria.

Additionally, an example-dependent pruning metric is defined by

$$PC_c = \text{Cost}(f(S)) - \text{Cost}(f^*(S))$$

where f^* denotes the classifier of the decision tree without the pruned node.

This cost-sensitive decision tree consists only of a single tree which holds the same negative properties as a regular decision tree, i.e., stability and overfitting. Creating an ensemble from these is demonstrated by (Bahnsen, Aouada, et al., 2015a) and outlined in the next section.

Ensembles of cost sensitive tree based classifiers

Bahnsen, Aouada, et al. (2015a) create random subsamples of the original training set S for T different base classifiers $j = 1, ..., T$ to fit an algorithm M_j per each subsample. In this case, M are cost-sensitive decision trees trained on different portions of the data. They offer several strategies for creating the subsets: bagging, pasting, random forests and random patches.

bagging randomly draws a bootstrap sample and then fits a base classifier. *pasting* uses random sampling without replacement. *random forests* use regular decision trees as base learner and internally apply bagging as well. The *random patches* approach applies random bootstrap sampling for both observations and features. Then, base classifiers created with one of the previously mentioned approaches are combined using either majority voting, cost-sensitive weighted voting, and cost-sensitive stacking.

Cost-sensitive weighted voting adds a weight α_j to each base classifier M_j during the voting phase (Breiman, 1996):

$$f_{wv}(S, M, \alpha) = \arg\max_{c \in \{0,1\}} \sum_{j=1}^{T} \alpha_j \mathbb{1}_c(M_j(S))$$

where $\mathbb{1}_c()$ is an indicator function which returns one if $c = 1$ or zero otherwise, and $\alpha = \{\alpha_j\}_{j=1}^{T}$ which in return is usually defined as the normalized misclassification error ϵ of the base classifier M_j in the out of bag set based on the set difference between the toal samples and the randomly chosen samples for this specific base classifier $S_j^{oob} = S - S_j$ resulting in:

$$\alpha_j = \frac{1 - \epsilon(M_j(S_j^{oob}))}{\sum_{j1=1}^{T} 1 - \epsilon(M_{j1}(S_{j1}^{oob}))}$$

which in return can be replaced by

$$\alpha_j = \frac{\text{Savings}(M_j(S_j^{oob}))}{\sum_{j1=1}^{T} \text{Savings}(M_{j1}(S_{j1}^{oob}))}$$

where *Savings*() is defined in Equation (3.4). Bahnsen, Aouada, et al. (2015a) explain that considering the real savings of the classifier better tackles real world problems.

cost-sensitive stacking

Stacking means fitting the second classifier on top of the output from a base learner (Hastie et al., 2001). Figure 3.33 shows a commonly seen setup of up to three levels of classifiers on Kaggle[8]. Kaggle is a data science competition platform where like-minded people from all over the world compete with statistical models to solve challenges. Base classifiers are weighted in the stacking procedure such that the ones which have greater savings are prioritized.

[8] `kaggle.com`

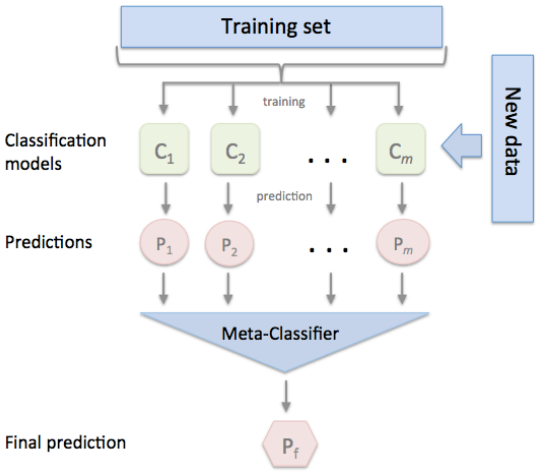


Figure 3.33: Demonstration of stacking multiple classifiers into an ensemble model (Raschka, 2016)

3.7 Model optimization

As the complexity of models grows, often the number of hyper-parameters to tune grows as well. When performing a systematic search for good hyper-parameters, historically grid search or random search was available. These require a lot of computation resources and not necessarily quickly generate good results.

Recently, Bayesian optimization gained traction to automate models. Several open source packages exist but might be tricky to run manually, and one cannot focus on the real problem to be solved.

We use sigopt[9] to optimize the model. Initially, a search grid is sent to the service. Then parameters are suggested. Cross-validated metric scoring results (mean, std) are returned to the service which in return suggests the next best parameters. The optimization loop continues until k observations are reached. In our experiments around 15-30 observations reported to the service already gave good results. The user experience was great, and it was good that we could focus on handling our problem domain and did not need to focus on implementing the Bayesian algorithms ourselves.

3.8 Cost matrix formulation for telecommunication industry

Following the proposed cost matrix by Bahnsen, Aouada, et al. (2015b) we define our own which suites the business needs of the telecommunication industry, see Table 3.1 for details.

[9] http://sigopt.com

Table 3.1: Cost matrix proposed for telecommunication industry to price individual risk.

	Actual Pos. ($y_i = 1$)	Actual Neg. ($y_i = 0$)
Predicted Pos. ($c_i = 1$)	$C_{TP_i} = 0$	$C_{FP_i} = r_i + C^m_{FP_i}$
Predicted Neg. ($c_i = 0$)	$C_{FN_i} = C_{device_i} + C_{marketInvest_i} + C_{usage_i} - D_i$	$C_{TN_i} = 0$

The motivation for us to introduce example-dependent cost is to

- be able to offer a fitting product to a customer with a slightly bad credit rating, i.e., not the latest iPhone but maybe the version from last year to reduce the initial upfront risk,
- have a method to handle the imbalance in the dataset,
- easier communicate results to the controlling department.

3.8.1 Correct classification

Vadera (2010) defines that is only sensible to assume that the cost of *correct* classification is smaller than misclassification, i.e., $C_{TN_i} < C_{FN_i}$ and $C_{TP_i} < C_{FP_i}$, which in our case are set to $C_{TN_i} = C_{TP_i} = 0$.

Next, we will have an in-depth look at the cost in case of erroneous predictions.

3.8.2 False positives

We define the cost of a false positive per customer C_{FP_i} as the sum of the opportunity financial cost and median risk cost, r_i and $C^m_{FP_i}$, where r_i describes the loss in profit if it had been a good customer. The profit per good customer r_i is calculated as the difference between the telecommunication provider's gains and expenses. We will not calculate the present value of money as initially proposed by Bahnsen as the controlling department is not applying these and the tax structure of our partner company is distorting the calculation with an interest rate of the group to the subsidiary which is not what a fair market value for a regular loan would have been. Additionally, revenue which could be obtained by selling possibly never paying customers to a factoring company which collects reminder fees and possibly returns parts of the initially invested money is not considered as the department told us that the returns are only marginal.

The difference between the telecommunication provider's gains and expenses is calculated given the cost of the device C_{device_i}, the minimal duration of the contract l_i, the $monthlyPayment_i$ plus the mean contract duration of a similar customer after initial

minimal contract duration defined as $\overline{l_{afterMinimumSimilar_k}}$ times the mean revenue per user of a similar user $\overline{r_{similar_k}}$ where k denotes the group of similar customers determined by *tariff segment, age group, partial zip code, isAustrian, sales channel, device price segment* and *deposit value segment*. We assume to at least see five customers per group. Otherwise, the overall mean is used as the default value.

$$r_i = (monthlPayment_i * l_i) - C_{device_i} + \overline{l_{\text{afterMinimumSimilar}_k}} \cdot \overline{r_{\text{similar}_k}}$$

The unsold device will not be kept in stock but rather sold to another customer instead. An alternative *average* customer must be assumed due to no prior knowledge median cost of the device median(C_{device_i}), median profit median(r) and median usage cost median(C_{usage_i}) are used to calculate the remaining risk as the cost originates from a long tail distribution. Usage risk is only considered in the case of problematic customers as it is assumed to be cared for by regular contract premiums:

$$C^m_{FP_i} = -\text{median}(r)\pi_0 + (\text{median}(C_{device}) + \text{median}(C_{usage})) \cdot \pi_1$$

In other words minus the profit of what would have been a good customer plus expected losses taking probability of default on first payment into account which is measured from the dataset, where $\pi_0 = 1 - \pi_1$,is calculated.

3.8.3 False negatives

We define the cost of a false negative per customer C_{FN_i} to be the losses if the customer never pays a single bill to cover the cost of the contract. Specifically, this is the value of the device C_{device_i} plus market invest $C_{marketInvest_i}$ minus the initial downpayment D_i if it exists. Additionally, as the current dunning process takes around three months due to legal obligations cost for unpaid usage of telecommunication services must be considered.

The cost of $C_{marketInvest_i}$ contains provision for the dealer, advertising cost subsidy for the dealer, overheads and the SIM card. Median usage cost median(C_{usage_k}) are incurred e.g. from roaming fees. As the usage cost is unknown at T_0 at the POS when the device is sold it is assumed to be similar to a peer group identified by *device value segment* and *tariff segment*. However, additionally one must also consider a higher usage cost which denotes the 75-percentile C_{u75p_k} which can be as large as the cost of a new device in extreme cases due to the nature of the long-tailed distribution.

$$C_{FN_i} = C_{device_i} + C_{marketInvest_i} - D_i + \pi_{mean} * Cu_k + \pi_{max} * C_{u75p_k}$$

where $\pi_{mean} = 1 - \pi_{max}$ and π_{max} is measured as the 75-percentile of the cost in the data. Note, for the calculation of usage related cost only the cost incurred by problematic, i.e., neverpayers is taken into consideration.

3.8.4 Handling of problems with the data

Several observations include problems with the original data. For example, the initial down payment is several times larger than the real value of the hardware sold to a customer. This distorts the notion of value at risk as there should not be a negative risk. Therefore, we use the *Median* of all correct cost formulations for C_{FP} and C_{FN} as a replacement value to make the risk more robust.

CHAPTER 4

Results

In the next sections, we review the distributions in the cost matrix, have a look at the result and discuss the outcome of different models. Subsequently, we will have a reflection about the strategy of the project, explainability of the model, technical debt for machine learning models, theft of the model when deployed, limitations of the savings based evaluation and the prototype as well as possible improvements concerning new data sources, feature engineering, and algorithmics.

4.1 Distribution of cost matrix

As the cost for correct classifications is set to 0, the cost for false positives and false negatives will be outlined in more detail here.

Figure 4.1 shows the cost for false positives by type of customer and 4.2 shows the same plot for cost of false negatives. As expected there are fewer cases of false positives than false negatives due to the highly imbalanced dataset. We can see in Figure 4.1 that the cost of falsely identifying a regular customer as a neverpayer tends to be larger and the reverse for the false negatives in 4.2 that neverpayers have a peak at around 1300 compared to normal customers at around 800.

We computed the ratio between the cost of false positives and false negatives. Figure 4.3 depicts the histogram by the target variable, i.e. normal and never paying customers, Figure 4.4 shows the log. The regular ratio tends to be a bit larger for neverpayers, but the differentiation is clearer in the log visualization of Figure 4.4. Here, we see that neverpayers have a larger ratio.

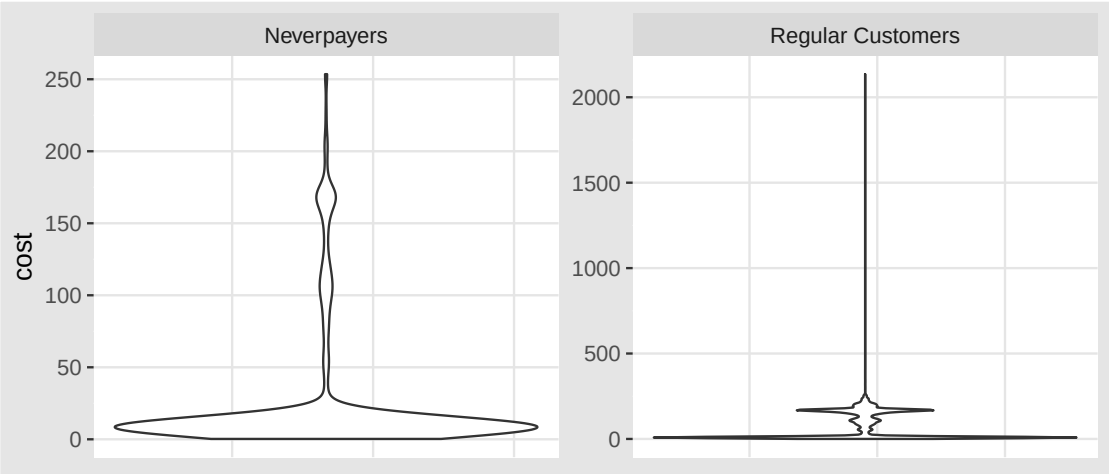


Figure 4.1: Cost for false positives per normal customers and neverpayers, respectively.

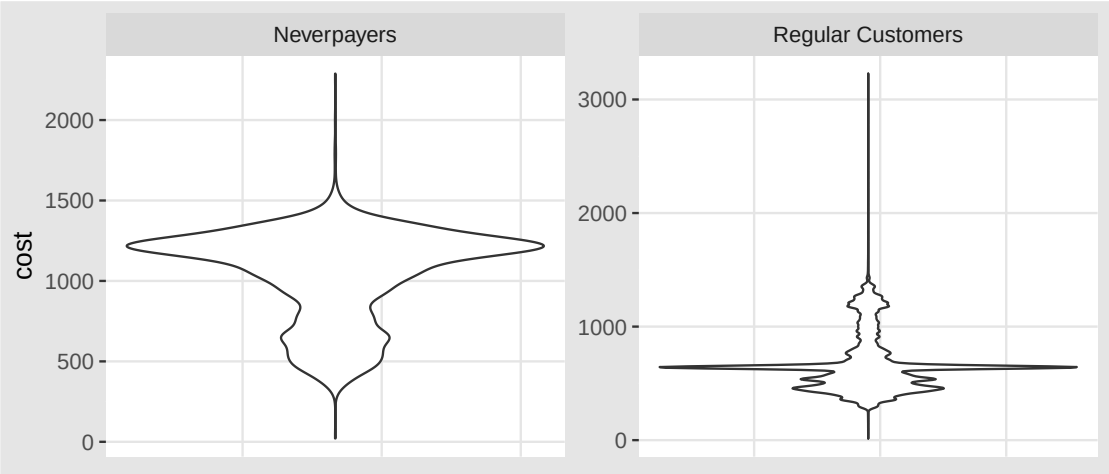


Figure 4.2: Cost for false negatives per normal customers and neverpayers, respectively

4.2 Comparison of models

In this section, we compare the new cost sensitive model with the currently applied credit scoring methodology at our partner and contrast it with special cost-sensitive algorithms by evaluating all models by F_2 score and the *savings* criterion.

As a reminder, the current approach is a traffic light system. To make the models comparable, we can only compare the automated part of the current credit check process and must ignore the manual actions. Therefore, there are two cases to differentiate:

- *red* predictions are assumed to be a neverpayer and *green* a regular customer (*TMA (current)*)

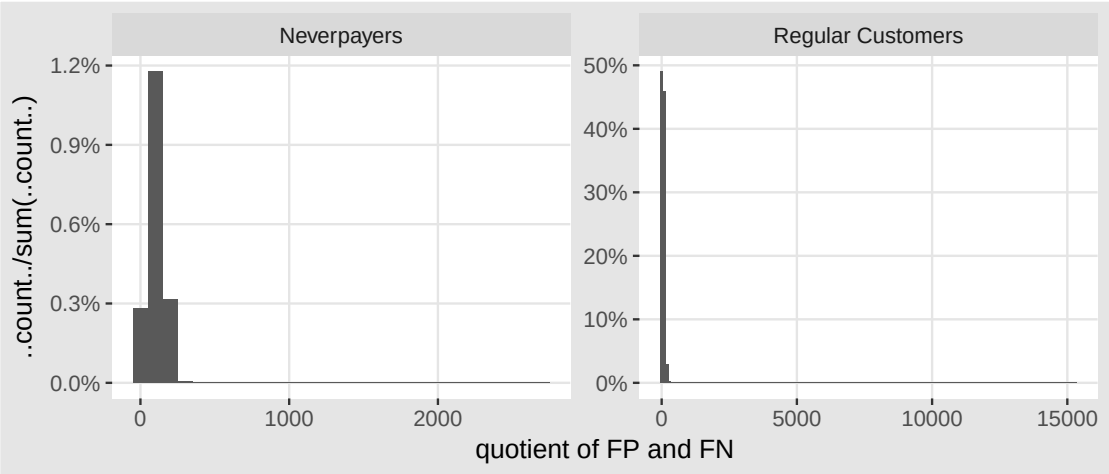


Figure 4.3: Cost coefficient per normal customers and neverpayers, respectively.

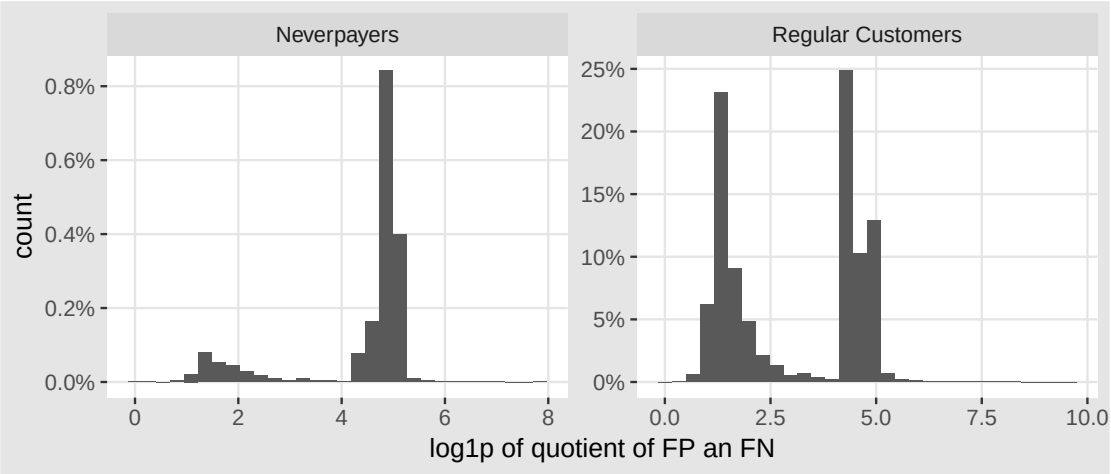


Figure 4.4: log of cost coefficient per normal customers and neverpayers, respectively.

- *red* and *yellow* predictions are assumed to be a neverpayer and *green* a regular customer (*TMA (current, assuming Yellow as neverpayer)*)

The goal for the department at our partner was to see if intelligent machine learning algorithms find at least 50% of the neverpayers per month and having a minimal error of false positives and false negatives. These targets (total & 50% of neverpayers) are outlined as horizontal lines in the next two plots. The lighter color outlines the false negatives, the darker one the true positives per month.

When comparing the regular cost based machine learning model with the current approach (*red* only) in Figure 4.5 we can see that the new model almost always fulfils this goal where the current approach is fairly conservative.

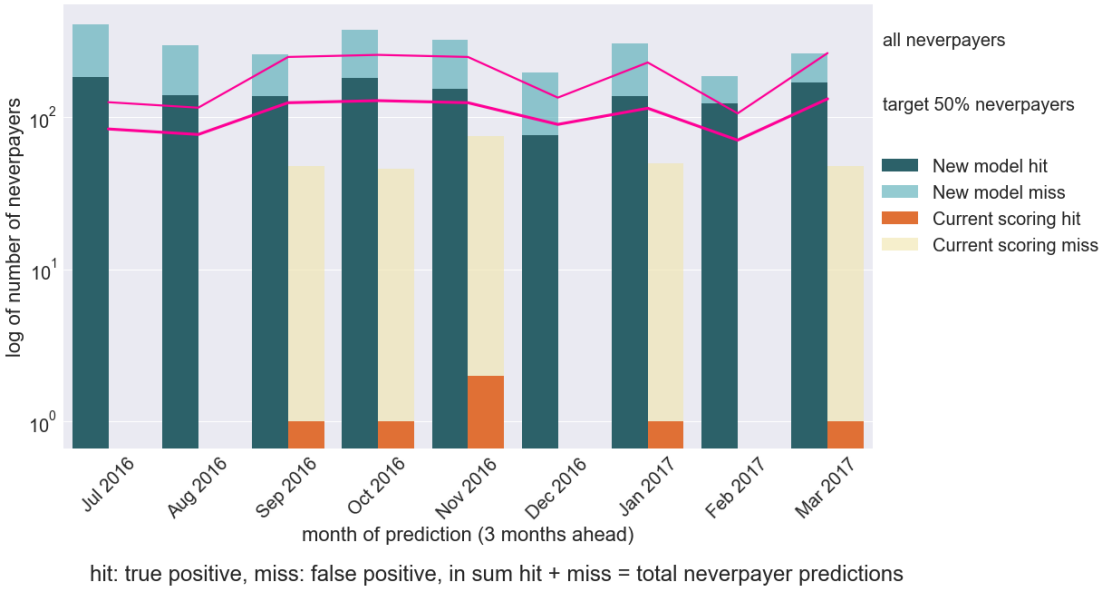


Figure 4.5: Classical ML model without savings compared with current approach *red*

In contrast Figure 4.6 outlines the second case (*red and yellow*). There, at least some neverpayers are always identified by the current approach, but one can see that a large number of customers is identified as a false negative. As mentioned before - this comparison is a bit hypothetical as the manual labour is not compared. However, one can see that the agents at our partner have to manually handle a vast number of cases for the current model to work out fine. Except for December, i.e., Christmas, the model seems to fulfill the goals well.

Having a closer look at the second case and the F_2 score 4.7 clearly shows the trend we identified before.

Even though the F_2 score is better for the current approach (*red and yellow*) when additionally computing the savings criterion as shown in Figure 4.8 it gets apparent that the current approach would be by far too aggressive.

To compare the different machine learning models we take the new gradient boosting approach as the benchmark and compare it with other classical and new cost-based algorithms.

Several classical machine learning models such as logistic regression, decision trees or a random forest are compared with the gradient boosting approach in Figure 4.9. However, we used multiple versions of the gradient boosting model to introduce the notion of cost and being able to compute the savings score. For the classical model, we see that all of these overall offer worse evaluation results. The input data was the same for all models, i.e., we did not specially create a dataset best suitable for logistic regression by normalizing numeric variables or removing collinearity. We assume that this could lead

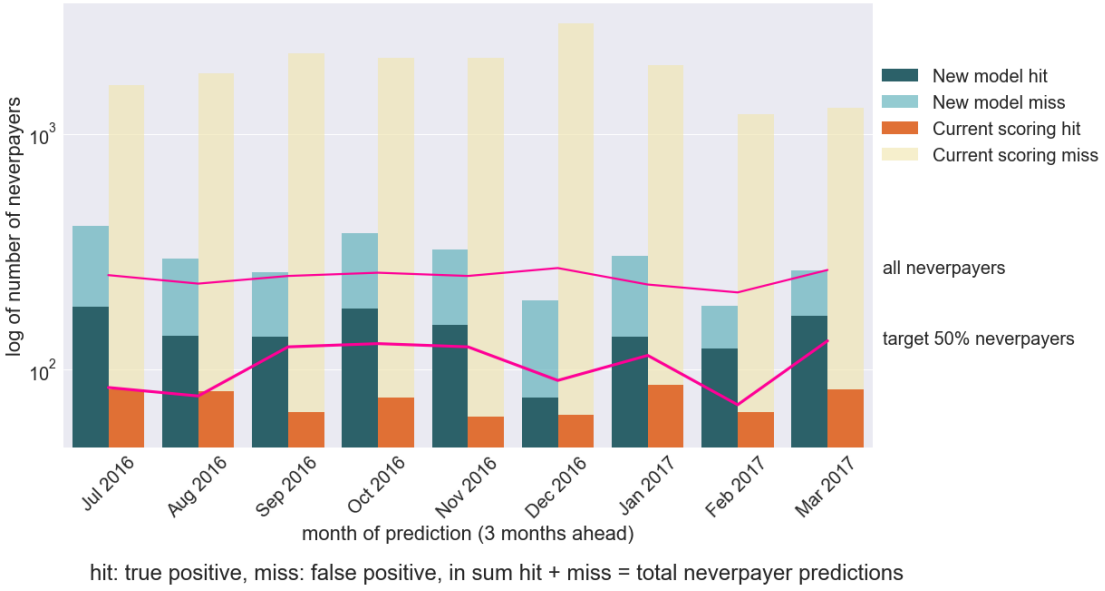


Figure 4.6: Classical ML model without savings compared with current approach *red and yellow*

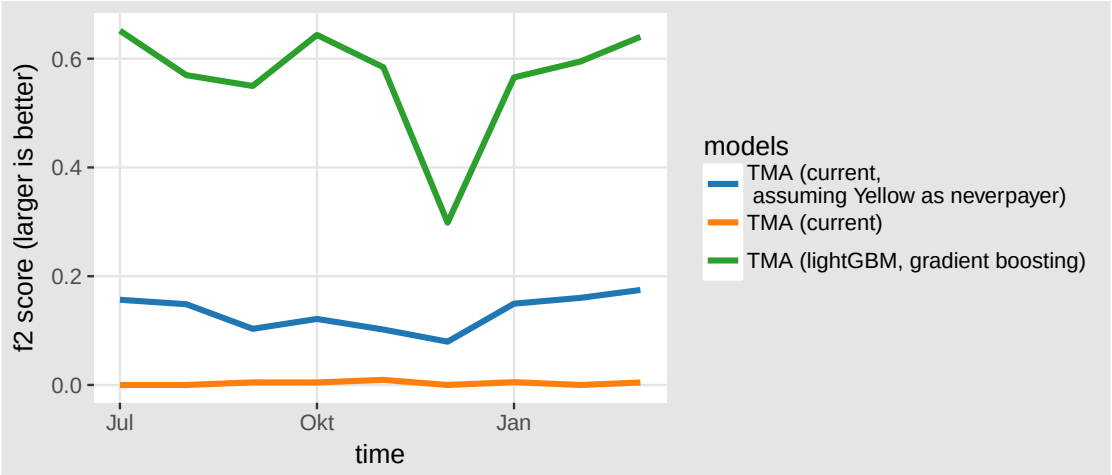


Figure 4.7: F2 compared with current approach *red and yellow*

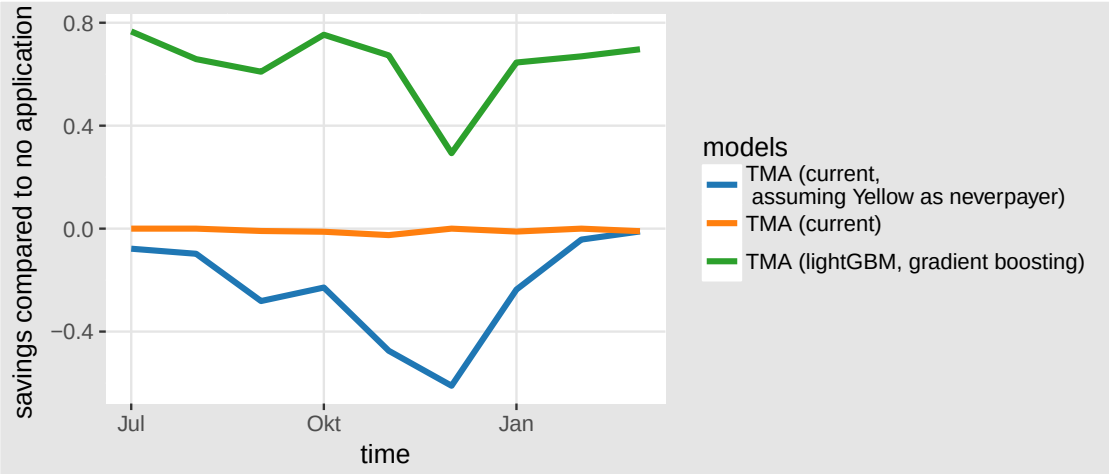


Figure 4.8: Savings compared with current approach *red and yellow*

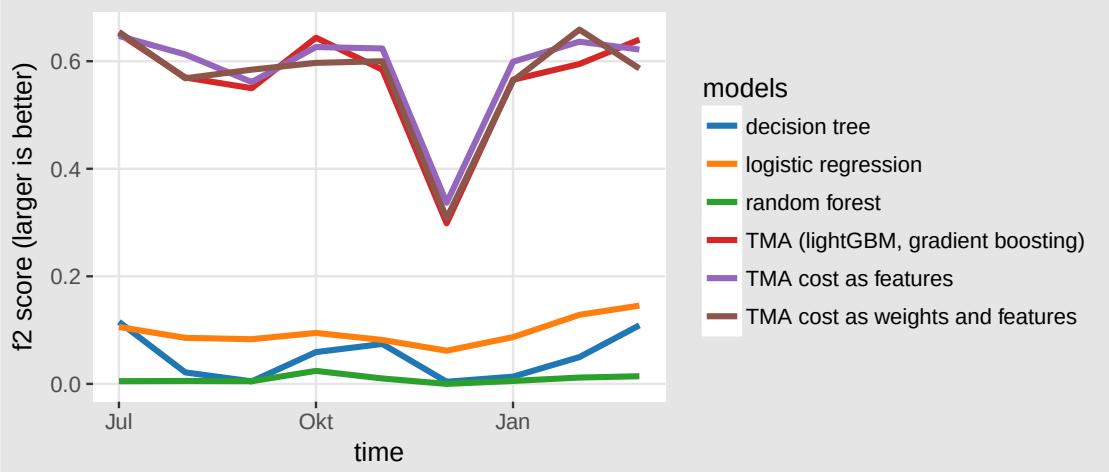


Figure 4.9: Comparing F_2 score over machine learning models (*cost might be present but no explicit cost-based algorithms*)

to an even better linear model.

Interestingly, decision trees and logistic regression seem to perform better than random forests, but when looking at the savings we see a similar problem as before in Figure 4.10 as well that a little bit higher F_2 score can translate to negative savings.

All three variants of the gradient boosting model have nearly similar evaluation results both for F_2 score and savings. Model *TMA cost as features* introduces the cost of false positives and false negatives as well as the cost and log based cost coefficient as features to the model, model *TMA cost as weights and features* additionally uses the weights parameter to pass an individual cost based weight to the gradient boosting algorithm.

A variety of cost based models is compared in Figure 4.11 by F_2 score and in Figure

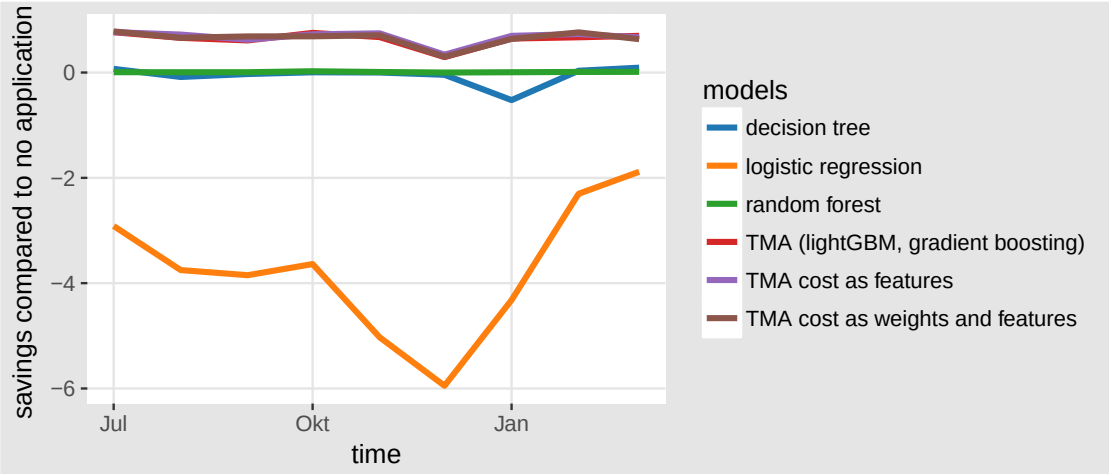


Figure 4.10: Comparing savings over machine learning models (*cost might be present but no explicit cost-based algorithms*)

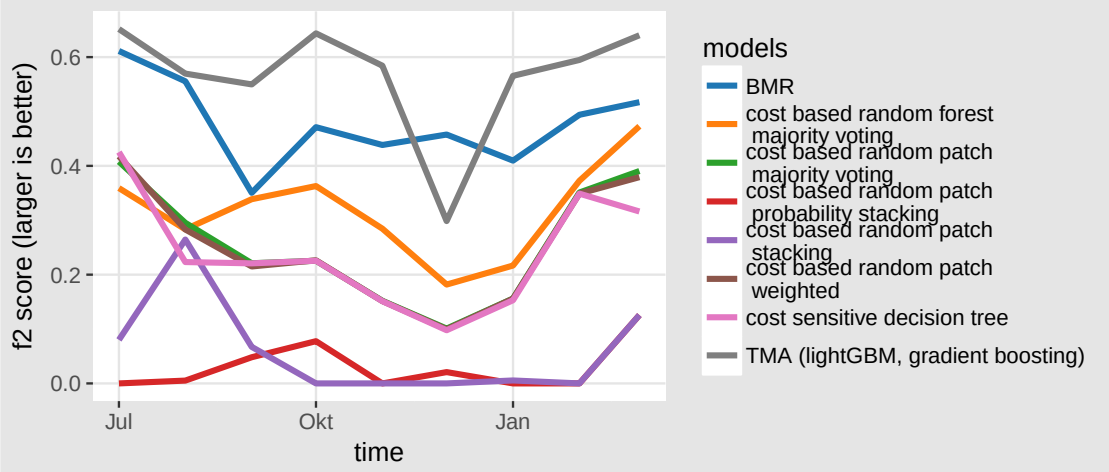


Figure 4.11: Comparing F_2 score over machine learning models (*cost-based algorithms*)

4.12 according to the savings criterion. For comparison, the gradient boosting model is included as well. Note, the BMR is based on the gradient boosting models output as probabilities and the entries to maximize the expected value. All other models have a lower F_2 score, but not necessarily a lot lower savings - though cost based random patches based on weighting or probability stacking have negative savings.

As this distorts the plots we have a look at the best performing cost based models in more detail in Figure 4.13 and 4.14. There we observe that the gradient boosting model with costs had the highest F_2 score but compared to BMR fewer savings. Interestingly the cost-sensitive decision tree tends to have a lower F_2 score but for the Christmas season provides more savings than the gradient boosting model with a higher F_2 score

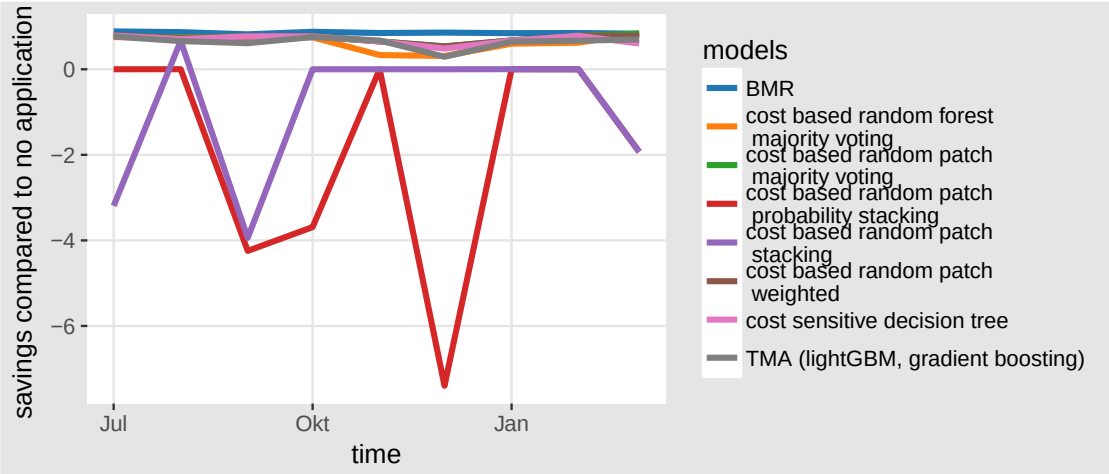


Figure 4.12: Comparing savings over machine learning models (*cost-based algorithms*)

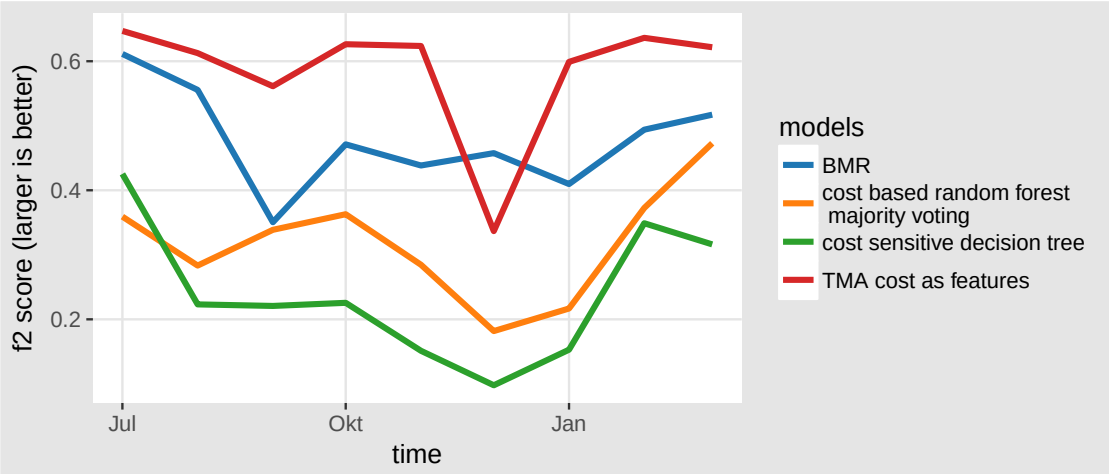


Figure 4.13: Comparing F_2 score over best models

for that month.

4.3 Critical reflection

Detection of fraudulent activities is an important activity in many industries. Especially for telecommunications where fraudulent behavior leads to the most significant loss of money. Digitalization of telecommunication leads to massive data sources which previously were not optimally used to prevent fraud, as often telecommunication providers relied on external credit scoring agencies using old and analog data and scoring methodologies.

As seen in Section 4.2, the regular credit check process provides neverpayer scoring

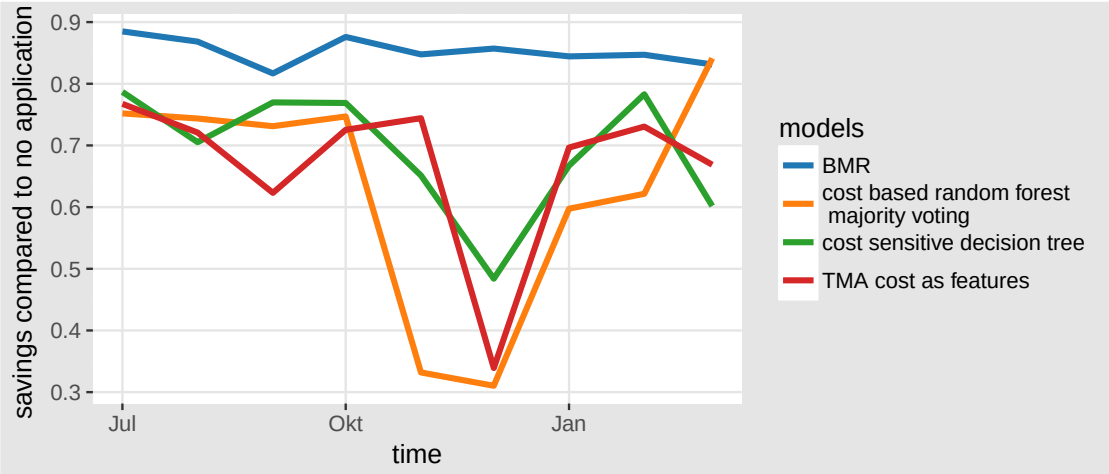


Figure 4.14: Comparing savings over best models

results which are good - but not due to the automated scoring, but due to the labor-intensive manual credit check process of our partner. We note that modern machine learning models can fulfill the expectations set by the department. Classically common models might produce reasonable classification results but can produce negative savings. Unique cost-based algorithms are not optimized for speed and might take a long time to compute. Interestingly, these did not give the best results in savings. However, BMR which combines the power of an input classifier which in our case is an industry grade gradient boosting model (*lightGBM*) with the expectation-maximization of the savings computed from the cost matrix yields the overall best savings.

Ethical and legal issues need to be looked at in detail if such a solution is deployed into a production system especially due to the upcoming additional regulation due to new privacy regulations (GDPR).

We could observe that nonrelative, absolute metrics, i.e., true positives, false positives are understood best by the departments as these can quickly be checked by hand by the business user in an excel sheet.

4.3.1 Observations about strategy

We observed that initially, the business did not have a good understanding of the evaluation process of machine learning models and which metrics to optimize for. Additionally, the data pipeline supposedly already in place when we were onboard turned out to be slightly unstable - especially when additions to features were required. Furthermore, after achieving the desired results of the business only then, we learned that some data sources used in the analysis could not be used in a real-time production setup which will lead to additional complexity for a production deployment as part of the data pipeline need to be reworked.

For a next data science project, we recommend having a reliable data pipeline in place before starting with exploration or model building. We recommend partnering the business needs from the department with an internal data scientist to make sure that the understanding about which metric should be optimized is aligned with the goals of the business from the start.

4.3.2 Explainability

In the past, simpler models were commonly used as these could be explained easier to the business or to a customer who was declined credit. Often we hear the argument that linear models, decision trees and sometimes even random forests are better understood than gradient boosting.

We want to point out: indeed, linear models might be simple, but only if applied directly to the data and not used on the principal components obtained by a principal component analysis. Decision trees are only more natural to explain when the depth is low and the number of features is not too high. For random forests, we do not see any easier explainability than for gradient boosted trees, especially as the only difference is the way how the training process is performed.

In fact, when looking at the plot in Figure 4.15, it is rather apparent that a linear classifier might not be the ideal tool to find the complex decision boundaries required in this context. The plot was produced by applying t-SNE (Van Der Maaten et al., 2008) to produce a two-dimensional embedding of our high dimensional dataset.

However, we believe that more complex models like gradient boosted trees or deep learning definitely can be used and explained well to business users or a customer when tools like lime[1] (Ribeiro et al., 2016) are used to transform the result into an easily and quickly understandable graphics.

Legal requirements are still unclear for automatic model decisions. It could be enough if an audit trail is available and the possibility of natural explanation through using such a tool is applied or if explicitly simple models, i.e., linear ones are mandatory.

A couple of months before the new *GDPR* regulation is applied, we believe that an audit trail which model was used to create a prediction is essential, but one must not resort to using simpler models due to the fear of not being able to explain them to customers.

Further investigation is required to understand the legal implications of switching from an externally provided credit score to an in-house prediction of credit worthiness according to similarity to a peer-group.

4.3.3 Technical debt

As pointed out by Sculley et al. (2014) the deployment of machine learning models quickly accumulates technical debt. To ease the deployment of future models and help reduce

[1] `https://github.com/marcotcr/lime`

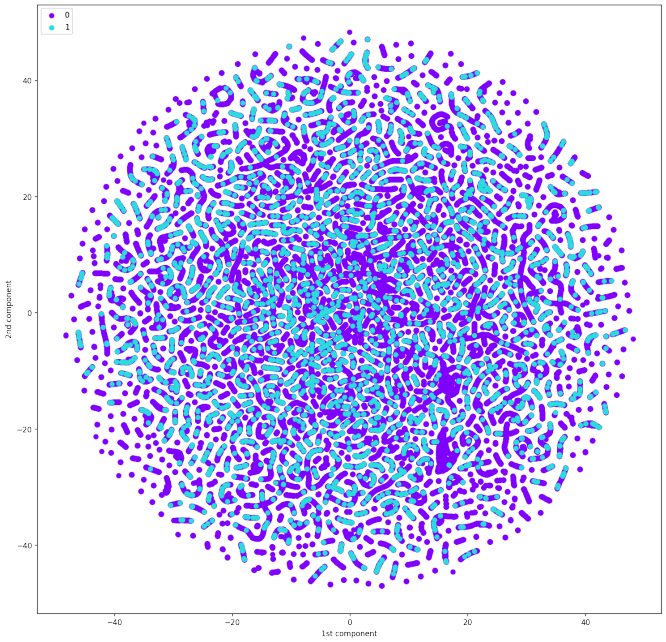


Figure 4.15: two-dimensional embedding produced by t-SNE of the data used in this thesis clearly shows that a linear classifier will have a hard time achieving great performance separating the groups.

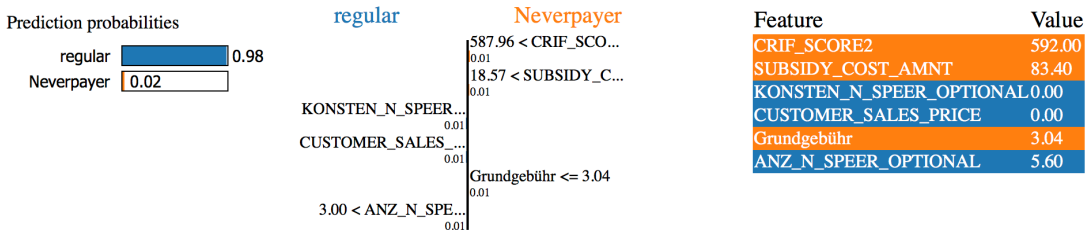


Figure 4.16: lime in action - easily explaining a complex model to the user for an individual observation in the dataset.

the debt or even better prevent it in the first place we suggest to implement a machine learning platform similar to michelangelo[2] and fbLearnerFlow[3] in-house engineered by *uber* and *facebook* as we expect that our partner will have multiple machine learning projects in the future in different national subsidiaries.

4.3.4 Stealing the model

Tramèr et al. (2016) point out how machine learning models deployed as an Application Programming Interface (API) can be reverse engineered by repeatedly querying the model with parameter sweeps to identify the decision boundaries. Interestingly, already the old model, i.e., the result obtained from the external credit check partner was abused by big electronic chains when selling other devices than our partner's hardware. To cope with the outlined problems one approach could be to protect the API of the model by only allowing up to a certain number of queries per user per time interval.

But also simpler scenarios need to be considered when looking at the upcoming privacy regulation. What about a fraudulent customer trying to brute force the model at the counter of real shops?

4.3.5 Limitations of prototype - suggestions for integration in a business process

During the prototype technical debt was accumulated by exploring a lot of feature engineering methodologies (Sculley et al., 2014). The data pipeline needs to be rebuilt from the ground up by using only data sources available for real-time access.

Additionally, the code needs to be improved, i.e., modularized and tests introduced to verify that the model behaves correctly after new features and improvements are added.

The prediction results need to be made available to the business - preferably via a API which easily can interface with existing business applications. As Juszczak et al. (2008) outlines Service Level Agreements (SLA) should be defined for uptime and response time of the model when it is integrated as an essential part of business processes.

The prototype assumes a clean ground truth dataset. In a real production, setup self-increasing feedback loops need to be prevented, and a strategy needs to be worked out how to deal with incomplete information as no valid ground truth information will be available when the model performs automated decisions. We recommend here to use a holdout group and forward $x\%$ of customers predicted as neverpayers still into the business process to have at least partial access to this data. Additionally, we suggest to initially use the output of the model as input for the manual credit check process to ease deployment as no automated business process needs to be changed.

[2] https://eng.uber.com/michelangelo/

[3] https://code.facebook.com/posts/1072626246134461/introducing-fblearner-flow-facebook-s-ai-backbone/

4.3.6 Limited usefulness of savings metric

In a production setup, we noticed that the savings metric is excellent to communicate results to controlling, i.e., as a relative metric of quality of the model. However, one needs to note, that due to not optimal quality of the data for some of the values required for cost calculation (i.e. NaN values) or insensible cost ratios (i.e. down payment larger than cost of device) or default values and the desire of the business to not merely take binary actions as well as the design of the savings metric, the values obtained cannot be considered absolute cost savings. The business would like to apply a *product based offering* to take fraud probability into marketing considerations and offer a suitable product to the individual customer and the individual financial situation. Therefore, one would not merely decline service on the prediction of the model but rather try to sell a different device or dynamically increase down payment. Furthermore, in the current setup, the decision process is not automated, i.e., the output of the model will not be used as the sole decision for the business process, rather it will be used as a tool to act as an input for decision support of existing manual credit check process.

4.3.7 Future improvements

Several possibilities to improve the model have been identified. New data sources can be added, new features computed or the model created in more complex ways.

Additionally, reinterpreting the classification problem as a regression on savings could be an exciting research project of its own. Also refining the cost matrix to account for (early) churn prediction or improving the cost calculations by adding the planned and currently remaining budget could be worth further research.

However, the model and all the features need to be maintained in a production setup. Therefore, we suggest a critical examination of each feature added if it is worth the increase in scoring metric to invest in its maintenance.

Data sources

New data sources like the web-shop (clickstream & device fingerprint) could be explored. Also, SAP or the shop system holds interesting data of which products are bought together. Sharing data with other national subsidiaries should be evaluated. More columns which contain information about the job of the applicant, family status or phone number portability could be worthwhile to examine.

Additional data can be analyzed in case of prediction as a neverpayer by applying usage data monitoring to perform anomaly detection or social network analysis or call behavior analysis.

Future research projects could try to use deep-learning to automatically identify modified documents and help the agents during the credit check process or check similarity of signatures to the signatures of fraudsters.

Feature engineering

Additional features could be computed. Among these, the distance between the customer and the next shop location according to geo-coordinates could be promising.

We already compute certain string distance metrics using fuzzyWuzzy[4], but additional phonetic similarity measures as outlined in jellyFish[5], could help to identify similar identities.

Collected e-mail addresses could be validated for correctness not only via regex but also by actually checking mailboxes for real existence of the supplied e-mail addresses.

Currently, only parts of the response from the external credit scoring agency are used. Several features, i.e., number of similar detected customers, and flagging of users into categories are already implemented on their side and could be applied by extracting appropriate features.

Additional features could be extracted by adding openly available data.

Bank account information could be checked for validity, i.e., *IBAN* provides a checksum which could be computed.

Algorithmics

Additional methods to handle categorical data as outlined in Soukhavong (2017) - especially binary coding should be experimented with.

Constantly new algorithms are introduced. Recently, infiniteBoost[6] & catBoost[7] were published. These could be validated.

Also automatic inference of machine learning pipelines as provided by tools like automl[8] or SuperLearner[9] could be worth further examination to also enable business users to utilize the power of machine learning.

Most importantly, different models can be ensembled to create more powerful models. This should be explored further, but deployment complexity needs to be taken into account. We suggest to only move further with ensembling after a fully fledged machine learning platform was set up. clipper[10] seems to be the most promising open source variant for such purposes.

A custom objective could be implemented to improve the cost-based evaluation of gradient boosting models. Currently, these internally still use non-cost-sensitive evaluation

[4] https://github.com/seatgeek/fuzzywuzzy
[5] https://github.com/jamesturk/jellyfish
[6] https://github.com/arogozhnikov/infiniteboost
[7] https://catboost.yandex
[8] http://www.ml4aad.org/automl/
[9] https://cran.r-project.org/web/packages/SuperLearner/index.html
[10] http://clipper.ai

metrics to optimize the loss function. Introducing such a custom objective would require the hessian and the gradient to be computed in a preferably convex, continuous finite and non-zero form which might require some changes to the current formulation of the cost matrix.

CHAPTER 5

Conclusion

We have shown that adding machine learning models to the data our partner is providing for credit scoring purposes can outperform the automated outcomes of their credit scoring process for neverpayer detection. Experiment outcomes show that the savings are highest when using an industrial grade machine learning classifier based on gradient boosting, *lightGBM* in our case, and combining it with an expectation maximization algorithm based on the cost matrix based on individual cost and probabilities using BMR. Too simple regular machine learning models, as well as other cost-sensitive classifiers, did not fare that well. Probably this can be attributed to the high imbalance of the data set and nonlinear decision boundaries.

For the business we suggest to bring smartness to their processes by integrating such machine learning models as a lot of the current credit scoring value is generated by the labor-intensive manual in-house credit check process. However, to do so successfully, the data needs to be available with sufficient quality in real time, which might pose a problem for some features due to legacy applications.

List of Figures

List of Tables

Acronyms

API Application Programming Interface. 69

BMR Bayes Minimum Risk. 50, 51, 64, 66, 73

CART classification and regression tree. 46–48

CDR Call Data Records. 8, 18

CRISP-DM Cross Industry Standard Process for Data Mining. 3, 14

CRM Customer Relationship Management System. 17–19

DSG Datenschutzgesetz. 12

DWH Data Ware House. 17, 19, 20

GIGO Garbage In Garbage Out. 33

IDS/IPS Intrusion Detection / Intrusion Prevention System. 9, 11

KPI Key Performance Indicator. 8, 18, 26

KYC Know Your Customer. 20

MVNO Mobile Virtual Network Operator. 22

PBX Private Branch Exchange. 6

POS Point of sale. 2, 16, 17, 34, 56

SLA Service Level Agreements. 69

SMOTE Synthetic Minority Over-sampling Technique. 37, 39

SOM Self Organizing Map. 7

SVM Support Vector Machine. 10, 11

TKG Federal Austrian Telecommunicaiton Act 2003. 2, 11, 12, 16

Bibliography

A. Lobe (2016). *Social-Media-Daten werden zur Ampel für die Kreditwürdigkeit.* URL: http://mobil.derstandard.at/2000042297313/Social-Media-Daten-werden-zur-Ampel-fuer-die-Kreditwuerdigkeit (visited on 08/05/2016).

Altini, M. (2015). *Dealing with Imbalanced Data: Under-Sampling, Over-Sampling and proper Cross-Validation.* URL: http://www.marcoaltini.com/blog/dealing-with-imbalanced-data-undersampling-oversampling-and-proper-cross-validation (visited on 08/07/2017).

Bahnsen, A. C., D. Aouada, and B. Ottersten (2014). Example-Dependent Cost-Sensitive Logistic Regression for Credit Scoring. *Proceedings - 2014 13th International Conference on Machine Learning and Applications*. Ed. by Xue-wen Chen, G. Qu, P. Angelov, C. Ferri, J.-h. Lai, and M. A. Wani, 263–269.

Bahnsen, A. C., D. Aouada, and B. Ottersten (2015a). Ensemble of Example-Dependent Cost-Sensitive Decision Trees. *Arxiv:1505.04637 [cs.LG]*.

Bahnsen, A. C., D. Aouada, and B. Ottersten (2015b). Example-Dependent Cost-Sensitive Decision Trees. *Expert Systems with Applications* 42 (19), 6609–6619.

Bahnsen, A. C., A. Stojanovic, D. Aouada, and B. Ottersten (2013). Cost Sensitive Credit Card Fraud Detection Using Bayes Minimum Risk. *Proceedings - 2013 12th International Conference on Machine Learning and Applications*. Ed. by M. A. Wani, G. Tecuci, M. Boicu, M. Kubat, T. M. Khoshgoftaar, and N. (Seliya. Vol. 1, 333–338.

Batista, G., A. Bazzan, and M. C. Monard (2003). Balancing Training Data for Automated Annotation of Keywords: A Case Study. *Proceedings of the second Brazilian Workshop on Bioinformatics*. Ed. by S. Lifschitz, F. Nalvo, J. Almeida, Pappas Jr. Georgios Joannis, and L. Ricardo. Vol. 3. 2, 35–43.

Becker, R. A., C. Volinsky, and A. R. Wilks (2010). Fraud Detection in Telecommunications: History and Lessons Learned. *Technometrics* 52 (1), 20–33.

Bhattacharyya, S., S. Jha, K. Tharakunnel, and J. C. Westland (2011). Data Mining for Credit Card Fraud: A Comparative Study. *Decision Support Systems* 50 (3), 602–613. arXiv: 1009.6119.

Bolton, R. J., D. J. Hand, F. Provost, L. Breiman, R. J. Bolton, and D. J. Hand (2002). Statistical Fraud Detection: A Review. *Statistical Science* 17 (3), 235–255.

Breiman, L. (1996). Bagging Predictors. *Machine Learning* 24 (2), 123–140.

— (2001). Random Forests. *Machine Learning* 45 (1), 5–32. arXiv: arXiv:1011.1669v3.

Cahill, M. H., D. Lambert, J. C. Pinheiro, and D. X. Sun (2004). Detecting Fraud in the Real World. *Computing Reviews* 45 (7), 913–930.

Carter, D. L. and J. G. Carter (2009). Intelligence-Led Policing. *Criminal Justice Policy Review* 20 (3), 310–325.

Chakrabarti, D. (2004). Autopart: Parameter-Free Graph Partitioning and Outlier Detection. *Knowledge Discovery in Databases: Pkdd 2004: 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Pisa, Italy, September 20-24, 2004. Proceedings*. Ed. by J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi. Berlin, Heidelberg: Springer Berlin Heidelberg, 112–124.

Chan, P. K. and S. J. Stolfo (1998). Toward Scalable Learning with Non-Uniform Class and Cost Distributions : A Case Study in Credit Card Fraud Detection. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*. Ed. by R. Agrawal, P. Stolorz, G. Piatetsky, and G. Chair, 164–168.

Chandola, V., A. Banerjee, and V. Kumar (2009). Anomaly Detection. *Acm Computing Surveys* 41 (3), 1–6.

Chapman, P., J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth (Aug. 2000). *Crisp-Dm 1.0 Step-by-Step Data Mining Guide*. Tech. rep. The CRISP-DM consortium.

Chawla, N. V., K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer (2002). SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research* 16, 321–357. arXiv: 1106.1813.

Chen, T. and C. Guestrin (2016). XGBoost: Reliable Large-Scale Tree Boosting System. *Arxiv*, 1–6. arXiv: 1603.02754.

Cohen, I. and M. Goldszmidt (2004). Properties and Benefits of Calibrated Classifiers. *Knowledge Discovery in Databases: Pkdd 2004: 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Pisa, Italy, September 20-24, 2004. Proceedings*. Ed. by J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi. Berlin, Heidelberg: Springer Berlin Heidelberg, 125–136.

Correa Bahnsen, A., D. Aouada, A. Stojanovic, and B. Ottersten (2015). Detecting Credit Card Fraud Using Periodic Features. *2015 IEEE 14th International Conference on Machine Learning and Applications*, In press.

Correa Bahnsen, A., D. Aouada, A. Stojanovic, and B. Ottersten (2016). Feature Engineering Strategies for Credit Card Fraud Detection. *Expert Systems with Applications* 51, 134–142.

Correa Bahnsen, A., A. Stojanovic, D. Aouada, and B. Ottersten (2014). Improving Credit Card Fraud Detection with Calibrated Probabilities. *Proceedings of the 2014 Siam International Conference on Data Mining*. Ed. by M. Zaki, Z. Obradovic, P. N. Tan, A. Banerjee, C. Kamath, and S. Parthasarathy. April 24-26, 2014. 677–685.

Criminisi, A. (2011). Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. *Foundations and Trends in Computer Graphics and Vision* 7 (2-3), 81–227.

Daskalaki, S., I. Kopanas, M. Goudara, and N. Avouris (2003). Data Mining for Decision Support on Customer Insolvency in Telecommunications Business. *European Journal of Operational Research* 145 (2), 239–255.

Deeb, A. E. (2017). *The Unreasonable Effectiveness of Random Forests*. URL: https://medium.com/rants-on-machine-learning/the-unreasonable-effectiveness-of-random-forests-f33c3ce28883 (visited on 08/11/2017).

Dheepa, V. and R. Dhanapal (2013). Hybrid Approach for Improvising Credit Card Fraud Detection Based on Collective Animal Behaviour and Svm. *Security in Computing and Communications: International Symposium, Sscc 2013, Mysore, India, August 22-24, 2013. Proceedings*. Ed. by S. M. Thampi, P. K. Atrey, C.-I. Fan, and G. M. Perez. Berlin, Heidelberg: Springer Berlin Heidelberg, 293–302.

Dorfleitner, G. and H. Jahnes (2014). What Factors Drive Personal Loan Fraud? Evidence from Germany. *Review of Managerial Science* 8 (1), 89–119.

Drummond, C. and R. Holte (2003). C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling Beats Over-Sampling. *Workshop on Learning from Imbalanced Datasets Ii*, 1–8.

Elitedatascience (2017). *How to handle Imbalanced Classes in Machine Learning*. URL: https://elitedatascience.com/imbalanced-classes (visited on 08/05/2017).

Elkan, C. (2001). The Foundations of Cost-Sensitive Learning. *Seventeenth International Joint Conference on Artificial Intelligence*, 973–978.

Estêvez, P. A., C. M. Held, and C. A. Perez (2006). Subscription Fraud Prevention in Telecommunications Using Fuzzy Rules and Neural Networks. *Expert Systems with Applications* 31 (2), 337–344.

Ferguson, A. G. (2012). Predictive Policing and Reasonable Suspicion. *Emory Law Journal* 259.

Ganganwar, V. (2012). An Overview of Classification Algorithms for Imbalanced Datasets. *International Journal of Emerging Technology and Advanced Engineering* 2 (4), 42–47.

García-Teodoro, P., J. Díaz-Verdejo, G. Maciá-Ferná Ndez, and E. Vá Zquez (2009). Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges. *Computers & Security* 28, 18–28.

Geith, A. (2006). „Künstliche Neuronale Netze zur Missbrauchserkennung in Mobilfunknetzen Auf Basis Von Verbindungsdaten.“ MA thesis. Wirtschaftsuniversität Wien.

Ghosh, J. K., M. Delampady, and A. Samanta (2006). „Bayesian Inference and Decision Theory“. *An Introduction to Bayesian Analysis. Theory and Methods*, 356.

Goldstein, M. and S. Uchida (Apr. 2016). A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data. *Plos One* 11 (4), 1–31.

Hartmann-Wendels, T., T. Mählmann, and T. Versen (2009). Determinants of Banks' Risk Exposure to New Account Fraud - Evidence from Germany. *Journal of Banking and Finance* 33 (2), 347–357.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning*. Springer series in statistics. New York, NY, USA: Springer New York Inc.

Hauge, M. V., M. D. Stevenson, D. K. Rossmo, and S. C. Le Comber (2016). Tagging Banksy: Using Geographic Profiling to Investigate a Modern Art Mystery. *Journal of Spatial Science*, 1–6.

Henecka, W. and M. Roughan (2015). Privacy-Preserving Fraud Detection Across Multiple Phone Record Databases. *IEEE Transactions on Dependable and Secure Computing* 12 (6), 640–651.

Hilas, C. S. and J. N. Sahalos (2005). User Profiling for Fraud Detection in Telecommunication Networks. *5th International Conference on Technology and Automation*, 382–387.

Hoath, P. (1998). Telecoms Fraud, the Gory Details. *Computer Fraud & Security* 1998 (1), 10–14.

Hollmén, J. (2000). „User Profiling and Classification for Fraud Detection in Mobile Communications Networks“. MA thesis. Department of Computer Science and Engineering Tietotekniikan osasto.

Japkowicz, N. (2000). The Class Imbalance Problem: Significance and Strategies. *In Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI*, 111–117.

Jeni, L. A., J. F. Cohn, and F. De la Torre (2013). Facing Imbalanced Data Recommendations for the Use of Performance Metrics. *ACII*, 245–251.

Jensen K. (2012). *CRISP-Dm Process Diagram*. URL: `https://commons.wikimedia.org/wiki/File:CRISP-DM%7B%5C_%7DProcess%7B%5C_%7DDiagram.png` (visited on 07/16/2016).

Jonge, E. de and M. van der Loo (2013). An Introduction to Data Cleaning with R. *Statistics Netherlands*, 53.

Juszczak, P., N. M. Adams, D. J. Hand, C. Whitrow, and D. J. Weston (2008). Off-the-Peg and Bespoke Classifiers for Fraud Detection. *Computational Statistics and Data Analysis* 52 (9), 4521–4532.

Kabari, L. G., D. Nuka Nanwin, and E. Uduak Nquoh (2015). Telecommunications Subscription Fraud Detection Using Artificial Neural Networks. *Transactions on Machine Learning and Artificial Intelligence* 3 (6).

Kappa (2007). *Wiley Encyclopedia of Clinical Trials*. John Wiley & Sons, Inc.

Kuhn, M. and K. Johnson (2013). *Applied Predictive Modeling*. New York, NY: Springer New York.

Liu, A. (2005). *Work Procedures for Empirical Research (Regression and Sem)*. URL: `http://www.researchmethods.org/step-by-step1.pdf` (visited on 07/16/2016).

— (2016). *Apache Spark Machine Learning Blueprints*. Packt Publishing, 17.

Mählmann, T. (2010). On the Correlation Between Fraud and Default Risk. *Zeitschrift Für Betriebswirtschaft*, 1–28.

Mahmoudi, N. and E. Duman (2015). Detecting Credit Card Fraud by Modified Fisher Discriminant Analysis. *Expert Systems with Applications* 42 (5), 2510–2516.

Metcalf, J. and K. Crawford (2016). Where Are Human Subjects in Big Data Research? the Emerging Ethics Divide. *Big Data and Society*, 1–34.

Monard, M. C. and G. Batista (2002). Learning with Skewed Class Distributions. *Advances in Logic, Artificial Intelligence and Robotics*, 173–180.

Moreau, Y., B. Preneel, P. Burge, J. Shawe-Taylor, C. Stoermann, and C. Cooke (1997). Novel Techniques for Fraud Detection in Mobile Telecommunication Networks. *ACTS Mobile Summit*.

Morozov, I. (2016). „Anomaly Detection in Financial Data by Using Machine Learning Methods“. PhD thesis. Hochschule für Angewandte Wissenschaften Hamburg.

Moudani, W. and F. Chakik (2013). Fraud Detection in Mobile Telecommunication. *Lecture Notes on Software Engineering* 1 (1), 75–79.

Olszewski, D. (2012). A Probabilistic Approach to Fraud Detection in Telecommunications. *Knowledge-Based Systems* 26 (February 2012), 246–258.

Oxford English Dictionary (2015). *Oxford English Dictionary Online*. URL: http://dictionary.oed.com.

Padmaja, T., N. Dhulipalla, R. Bapi, and P. Krishna (2007). Unbalanced Data Classification Using Extreme Outlier Elimination and Sampling Techniques for Fraud Detection. *15th International Conference on Advanced Computing and Communications (ADCOM 2007)*, 511–516.

Patcha, A. and J. M. Park (2007). An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends. *Computer Networks* 51 (12), 3448–3470.

Pernul, G., P. Y. A. Ryan, and E. Weippl (2015). Learning from Others: User Anomaly Detection Using Anomalous Samples from Other Users. *Computer Security – Esorics 2015. Lecture Notes in Computer Science* 9327. Ed. by W. E. Pernul G. Y A Ryan P., 396–414.

Provost, F. and T. Fawcett (1997). Analysis and Visualization of Classifier Performance: Comparison Under Imprecise Class and Cost Distributions. *Kdd-97 Proceedings* (July), 43–48.

Raschka, S. (2015). *Python Machine Learning Essentials*. arXiv: arXiv:1011.1669v3.

— (2016). *Mlxtend*. URL: https://rasbt.github.io/mlxtend/user%7B%5C_%7Dguide/classifier/StackingClassifier/%20http://dx.doi.org/10.5281/zenodo.594432%20http://rasbt.github.io/mlxtend/ (visited on 11/11/2017).

Rbx (2014). *Kappa Statistic in Plain English*. URL: http://stats.stackexchange.com/questions/82162/kappa-statistic-in-plain-english (visited on 07/20/2016).

Ribeiro, M. T., S. Singh, and C. Guestrin (2016). Why Should I Trust You? Explaining the Predictions of Any Classifier. *Arxiv:1602.04938 [cs.LG]*. arXiv: 1602.04938.

Said Bleik, S. G. (2016). *Computing Classification Evaluation Metrics in R*. URL: http://blog.revolutionanalytics.com/2016/03/com%7B%5C_%7Dclass%7B%5C_%7Deval%7B%5C_%7Dmetrics%7B%5C_%7Dr.html (visited on 07/20/2016).

Sculley, D., G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, and M. Young (2014). Machine Learning : The High-Interest Credit Card of Technical

Debt. *Nips 2014 Workshop on Software Engineering for Machine Learning (SE4ML)*, 1–9. arXiv: arXiv:1011.1669v3.

Shawe-Taylor, J. (1999). Detection of Fraud in Mobile Telecommunications. *Information Security Technical Report* 4 (1), 16–28.

Soukhavong, D. (2017). *Visiting: Categorical Features and Encoding in Decision Trees*. URL: https://medium.com/data-design/visiting-categorical-features-and-encoding-in-decision-trees-53400fa65931 (visited on 08/05/2017).

Subudhi, S. and S. Panigrahi (2015). Quarter-Sphere Support Vector Machine for Fraud Detection in Mobile Telecommunication Networks. *Procedia Computer Science* 48 (Supplement C). International Conference on Computer, Communication and Convergence (ICCC 2015), 353–359.

T-Mobile Austria (2016). T-Mobile Austria Internal.

Taniguchi, M., M. Haft, J. Hollmen, and V. Tresp (1998). Fraud Detection in Communication Networks Using Neural and Probabilistic Methods. *IEEE International Conference on Acoustics, Speech and Signal Processing* 2, 1241–1244 vol.2.

Tramèr, F., F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart (2016). Stealing Machine Learning Models Via Prediction APIs. (Ml), 19. arXiv: 1609.02943.

UCLA IDRE Institute for Digital Research and Education (2014). *R Library : Contrast Coding Systems for Categorical Variables*. URL: https://stats.idre.ucla.edu/r/library/r-library-contrast-coding-systems-for-categorical-variables/ (visited on 11/11/2017).

Underwood, J. (2016). *Predictive Model Data Prep: An Art and Science*. URL: http://www.jenunderwood.com/2016/07/15/predictive-model-data-preparation-art-science/ (visited on 07/16/2016).

Vadera, S. (2010). CSNL: A Cost-Sensitive Non-Linear Decision Tree Algorithm. *Acm Transactions on Knowledge Discovery from Data* 4 (2), 1–25.

Van Der Maaten, L. J. P. and G. E. Hinton (2008). Visualizing High-Dimensional Data Using T-Sne. *Journal of Machine Learning Research* 9, 2579–2605. arXiv: 1307.1662.

Van Vlasselaer, V., C. Bravo, O. Caelen, T. Eliassi-Rad, L. Akoglu, M. Snoeck, and B. Baesens (2015). APATE: A Novel Approach for Automated Credit Card Transaction Fraud Detection Using Network-Based Extensions. *Decision Support Systems* 75, 38–48.

Vida, A. (2016). *Practical Guide to Deal with Imbalanced Classification Problems in R*. imbalanceAnalyticsVida. URL: https://www.analyticsvidhya.com/blog/2016/03/practical-guide-deal-imbalanced-classification-problems/ (visited on 08/07/2017).

Vrat, B., N. Aggarwal, and S. Venkatesan (2015). Anomaly Detection in Ipv4 and Ipv6 Networks Using Machine Learning. *India Conference (INDICON), 2015 Annual Ieee*. IEEE, 1–6.

Walber (2017). *File:Precisionrecall.svg - Wikimedia Commons*. URL: https://commons.wikimedia.org/wiki/File:Precisionrecall.svg (visited on 08/07/2017).

Wang, S. (2010). A Comprehensive Survey of Data Mining-Based Accounting-Fraud Detection Research. *2010 International Conference on Intelligent Computation Technology and Automation, Icicta 2010* 1, 50–53. arXiv: 1009.6119.

West, J. and M. Bhattacharya (2016). Intelligent Financial Fraud Detection: A Comprehensive Review. *Computers and Security* 57, 47–66.

Whitrow, C., D. J. Hand, P. Juszczak, D. Weston, and N. M. Adams (2009). Transaction Aggregation as a Strategy for Credit Card Fraud Detection. *Data Mining and Knowledge Discovery* 18 (1), 30–55.

Wieland, K. (2004). The Last Taboo? Revenue Leakage Continues to Hamper the Telecom Industry. *Telecommunications (International Edition)* 38, 10–11.

Wilcox, K. and A. T. Stephen (2013). Are Close Friends the Enemy? Online Social Networks, Self-Esteem, and Self-Control. *Journal of Consumer Research* 40 (1), 90–103. arXiv: arXiv:1011.1669v3.

Xing, D. and M. Girolami (2007). Employing Latent Dirichlet Allocation for Fraud Detection in Telecommunications. *Pattern Recognition Letters* 28 (13), 1727–1734.

You, W. and N. To (2016). Quantifying the Case for Enhanced Data Preparation | Blue Hill Research. (February).

Zadrozny, B., J. Langford, and N. Abe (2003). Cost-Sensitive Learning by Cost-Proportionate Example Weighting. *Third IEEE International Conference on Data Mining*, 435–442.