Improving Customer Experience - Predictive Model of Billing Service Requests

Pedro Barros de Oliveira

Master's Dissertation

Supervisor: Prof. Gabriela Beirão

Mestrado Integrado em Engenharia Industrial e Gestão

2019-07-01

Abstract

In a extremely competitive market where customer satisfaction is the main factor of loyalty and churn, the telecommunication companies always need to be focused on maintaining their clients as satisfied as possible with the service provided. One factor negatively impacting customer satisfaction is the complexity of the bills.

This dissertation focuses on increasing customer satisfaction levels by diminishing their problems related to billing. To address this problem, a predictive model of customer billing service requests was developed, using the customers' portfolio and characteristics. The aim is to identify which customers are more likely to contact a company due to their bill. The model developed employs machine learning techniques, using several algorithms, tree based, linear and neural networks. These models were evaluated regarding their overall performance in estimating clients' service requests as well as their business applicability.

The objectives are to reduce the number of billing service requests presented by the clients each month, to find a way of identifying clients' most frequent issues with bills and, consequently, deliver them the message in a clearer way, and finally to help the company manage their customer service resources more evenly.

Melhorar a experiência do consumidor - Modelo Preditivo de Dúvidas de Faturação

Resumo

Num mercado extremamente competitivo onde a satisfação do consumidor é o principal fator de lealdade e abandono, as empresas de telecomunicações precisam de se focar em manter os seus clientes tão satisfeitos quanto possível com os serviços proporcionados. Um dos fatores que impacta negativamente a satisfação do cliente é a complexidade das faturas que lhe são apresentadas.

O foco desta dissertação é na subida dos níveis de satisfação do cliente, diminuindo os seus problemas relacionados com faturação. Para resolver este problema, um modelo preditivo de dúvidas de clientes foi desenvolvido, usando as caraterísticas de portfólio dos mesmos. O objetivo é identificar quais são os clientes mais propícios a entrar em contacto com a empresa devido à sua fatura. Foram aplicadas técnicas de *Machine Learning*, usando diversos algoritmos distintos, algoritmos de árvore, regressões lineares e redes neuronais. Os modelos foram avaliados através da sua performance ao longo de todos os dados assim como a viabilidade da sua aplicação em ambiente empresarial.

Os objetivos propostos são reduzir o número de dúvidas de faturação apresentadas pelos clientes todos os meses, identificar quais são os probelmas mais frequentes dos clientes com as faturas, consequentemente apresentando as mesmas de forma mais clara e, por último, ajudar a empresa a gerir os seus recursos de apoio ao cliente mais homogeneamente.

Acknowledgments

This dissertation marks the conclusion of a five year period of college with a lot of good memories and hard work. I would like to firstly thank the partner company for allowing me to be a part of their team for the past six months, as well as my coordinators Carlos Pereira and Diogo Santos for helping me along the way and teaching me a lot.

I would also like to thank Prof. Gabriela Beirão, my supervisor for this master's dissertation, for helping me develop this thesis and for always answering my questions promptly. To my colleagues both at FEUP and at the partner company, I would also like to acknowledge them, for helping me along the way and especially for all the fun moments provided.

Lastly, a special thanks to my close friends, for always being there for me in both the good and the bad times, and to my family, for reminding me of what is important and constantly leading me to success.

"I could either watch it happen or be a part of it."

Elon Musk

Contents

Acronyms and Symbols

AUC	Area Under Curve
CRISP-DM	Cross-industry Standard Process for data mining
FE	Feature Engineering
FN	False Negatives
FP	False Positives
GB	Gradient Boosting
GLM	Generalized Linear Model
KDD	Knowledge Discovery in Databases
MCC	Matthews Correlation Coefficient
NN	Neural Networks
NPS	Net Promoter Score
RF	Random Forest
ROC	Receiver Operating Characteristic
SR	Service Request
TN	True Negatives
TP	True Positives
XGBoost	Extreme Gradient Boost

List of Figures

List of Tables

Chapter 1

Introduction

The background and motivation for the current project as well as the main objectives and methodologies used will be briefly mentioned in this chapter. The structure of the dissertation, including the description of each chapter and what to except is also present.

1.1 Project Background and Motivation

In the Portuguese telecommunications market, services are offered in different ways, with combinations of five main products, mobile and fixed network, mobile and fixed phone and television, available via bundles, as well as several extras like mobile data, premium channels, movies and others. There are also frequent campaigns, promoted by the companies, that offer customers a variety of discounts. All of these factors increase the complexity of the bills, making it common for customers to have doubts about what they are being charged for. In a time where customer satisfaction is viewed as one of the most important performance measures for a company, every factor that leads to discomfort for the client is viewed as a problem, as well as an improvement point, specially in a market as competitive as this one, where customer satisfaction can make a difference when choosing a provider.

The Portuguese telecommunications market is composed by four main players, NOS, Altice, Vodafone and Nowo, having a combined EBITDA (Earnings Before Interest, Taxes, Depreciation and Amortization) of around 1 989 billion by the end of 2017, with a turnover of 4 865 billion and a total of 25,4 million Revenue Generating Units.

With the companies closing the gap between each other in terms of the service and price provided, they start to distinguish themselves in terms of the quality of the product as well as the effectiveness of the customer relationship departments, which means that for most of these companies, customer satisfaction is a first level performance indicator. In an attempt to answer all of the customer complaints and questions, they have created big departments where hundreds of workers are in charge of receiving and analyzing customer service requests (SR). These departments receive millions of SR each year that, on one hand, leads to huge costs for the company and on the other, is a sign of generalized customer dissatisfaction.

Ideally, big companies should be able to predict any type of problem a customer will have, before they even know that they have it, this is where they can gather the most value from. In recent years, the expectations of clients have been shifting, a quick response to problems is becoming the norm and it is increasingly expected that the companies can predict certain problems and know the needs of the client beforehand. Predictive modeling, the technique this thesis is based around, is becoming immensely useful for companies because it can help them achieve just that.

This project was developed in one of the biggest telecommunication companies in Portugal, from now on referred as partner company, where there is an increasingly big effort in developing these techniques. They are now starting to apply these procedures to customer behaviour, in the case of this project, billing service requests, a very sensitive topic in the customers' perspective since it involves money. It is in this context that the project was developed, the creation of a predictive model of billing service requests. During this project customer data was always anonymized in order to ensure the protection of the customers' privacy.

1.2 Project Objectives

The project developed in this dissertation aims to predict, with high certainty, what clients are more likely to present service requests about their bills to one of the several customer supports centers of the partner company. The model was developed using the customers' characteristics as well as their past behaviour and also events that occurred during the billing period, for example, the service going down for a couple of days. Machine learning techniques were used for data mining purposes in a way to find relevant interactions for the project under development. The project emerges from a constant difficulty that telecommunication companies have in presenting clear bills to their customers, since there is a very low standardization and a very large volume of clients, in the case of the partner company, it has around 1,4M monthly bills to deliver.

Every year, the partner company receives around one million service requests just regarding billing problems, by more than 30% of its customer base. Assuming also that several clients have doubts but do not call, there is a serious problem that directly affects one of the most sensitive company areas nowadays, customer satisfaction.

The goal of this project is to develop a prediction model that allows the detection of the clients most likely to present a billing SR. Secondly, after predicting the clients most likely to contact the customer services of the partner company, it would also be useful to understand why are they contacting. By the end of the project, an algorithm was developed that allows the partner company to identify which clients are more likely to utilize one of its customer service centers so that a more advantageous solution for both parties can be found. For this project, a billing service request was considered any interaction between a client and the customer relations departments of the partner company that falls under the generalized typification of Billing.

1.3 Project Methodology

The project in hand can be split into three main phases, first of all, integration in the company, learning the project and studying the data. This phase included the presentation of all the members involved in data science activities at the partner company as well as an explanation of the context of the problem, what is its aim and what lead to its necessity.

The second and biggest phase of the problem was the construction of the dataset and the development of the prediction algorithm. A huge amount of work was put into collecting and creating relevant information for the model, with the majority of the most important variables coming from feature engineering. Different models were tested using distinct methods of data balancing, considering that the dataset had around 3% of the positive class.

Lastly, there was the selection of the best model and validation of the results. The model was chosen by its general performance, capacity of handling unbalanced data as well as its business applicability, several different performance metrics were utilized for comparing the performance that will be described in the following chapter. A chronograph with the activities developed during the project can be seen in Figure 1.1.

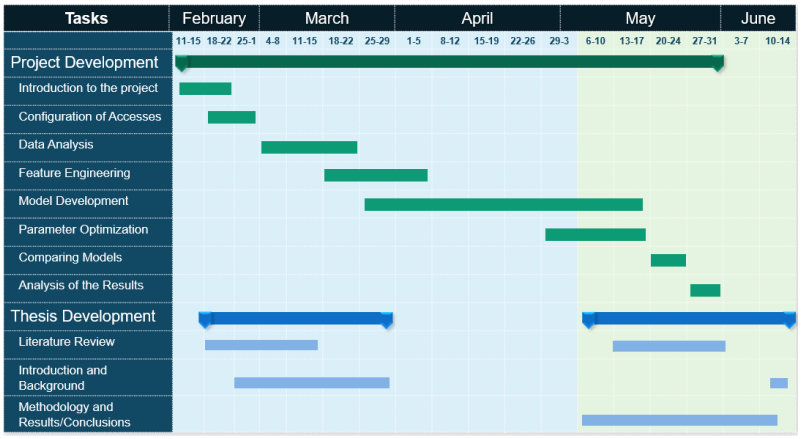


Figure 1.1: Chronograph of the project

1.4 Dissertation Structure

The present dissertation is split into six chapters as well as three annexes to help its comprehension. The present chapter contains a description of the project and its context. The methodology and objectives are also explained. The second chapter contains the state of the art covering the most important concepts for the understanding of this project, such as the prediction models, performance indicators and relevant information regarding prediction models. Also covers similar studies performed by other researchers and the gap this dissertation aims to fill.

The third chapter contains a more detailed explanation of the context of the problem and all the circumstances surrounding it. There is also an in depth explanation of the initial situation, the numbers surrounding it and how the problem is currently handled. Chapter three, also provides a

detailed description of the service requests, some statistics that help framing the situation, and the opportunities for improvement that the project will address.

The fourth chapter describes the methodology, detailing all the steps undertaken, such as the data mining procedures, the hurdles of the problem and the decisions taken in each step.

The results are presented in the fifth chapter and the different models compared in terms of performance and applicability. Lastly, in the final chapter, all the conclusions derived from the project will be enumerated. Future projects that can be developed in order to go deeper into this topic are also mentioned.

Chapter 2

Literature Review

The present chapter starts with a brief review of similar studies and an explanation of the project's relevance. Data mining procedures, including the different phases of the process, several techniques and also the different models and evaluation metrics used throughout the project are also described.

2.1 Customer complaints in the telecommunication context

The current research studies the possibility of predicting customer billing service requests in a telecommunications company. The bases for this analysis is the extremely high volume of questions and complaints a telecommunications company receives every month, which generate not only enormous costs for the companies, but also customer dissatisfaction.

Improving customer experience has become one of the main short term goals of big companies, this comes from the fact that, nowadays, customers interact with the companies through several different channels and media, resulting in more complex journeys and relationships. Moreover, the rise of social medias elevated the number of customer to customer interactions to extreme values and, consequently, new challenges and opportunities for the corporations appear. (Lemon and Verhoef, 2016)

Many statistical studies have been made in order to explore customer complaints, in an attempt to understand if they related with customer churn and what factors potentiate it the most. It was concluded by Bolton and Bronkhorst (1995) that customer complaints in telecommunication companies do in fact influence customer churn and doing things right the first time is critical for customers satisfaction. Previous research, developed by Schwartz and Overton (1987), showed that billing is a major source of customer dissatisfaction. Extant scientific research showed that customer satisfaction is very important for customer retention and that, in the telecommunication market, customer retention determines the success and chance of survivability of a company (Almossawi, 2012). Furthermore, in the eyes of some customers, the customer relationship service of a telecommunication company should be considered a core part of the business, in fact, some

telecommunications companies even consider it more important than the core service itself (Roos and Edvardsson, 2008).

There were also some attempts of using computational models in order to help with these topics. A computer program was developed by Sloo (1999), with the intent of automatically handling and resolving user complaints and have a faster and better solution to this problem. However, it is important to avoid customer churn and, another program intending to predict it using data from previous complaints, among other information, was developed (Hadden et al., 2006).

Although previous research has studied the implications of complaints and ways of optimizing the handling process, none has attempted to find a way of pro-actively predicting and preventing them, and this project aims to fill this gap. Furthermore, although the majority of studies only use data about complaints, this project aims to predict customer billing service requests, including not only complains, but also simple questions that a customer might have.

2.2 Knowledge Discovery in Databases

Knowledge discovery in databases (KDD), according to Fayyad et al. (1996), is the non-trivial process of identifying valid, potentially useful and ultimately understandable patterns in data. It starts with the data in its initial raw and unstructured way and ends when something of value is found. The KDD process involves several steps, as shown in Figure 2.1 and described as follows:

1. Selecting the dataset: Selecting a relevant dataset for the problem in hand, preferably with a significant amount of entries, so that the obtained results can have any significance. A study by Oreski et al. (2017) highlights the importance of the dataset's characteristics on the accuracy and time consumption of feature selection techniques.

2. Preprocessing the data: Rarely does the selected dataset come clean and ready for analysis, it is necessary to treat its outliers, missing values and noise. Outliers are very extreme values within the data that should be removed because even if they are true, they are a very unlikely event that will probably not happen again and can skew the data. An outlier can be detected if a value is more than three standard deviations above or below the average value for a certain variable. Missing values are entries in the dataset with missing variables, in these cases, several procedures can be done. A variable where most of the data is missing, should be removed from the analysis, if it is a variable with only a small percentage of entries with missing values, than those lines should be removed. In other situations, the best solution is to replace the missing values with the most probable value using a regression or the median. Noise are impossible values, for example, negative ages. These values should be removed or replaced with the most probable one, since they will always influence the model in a negative way. Preprocessing the data can lead to a better performance of the model, ensuring that all variables receive equal treatment during the training process (Wu et al., 2009).

3. Data reduction and normalization: Removing unnecessary columns and keeping only the ones that best explain the data. Often there are repeated columns or extremely correlated ones that add nothing to the problem and can be removed. It is not obligatory for the analyst to remove these columns since if they are not 100% correlated, they can be positive for the model. As such, the analyst needs to decide the desired trade-off between time and results. Data normalization can be important for certain problems, since they can have a big influence in the results. Datasets with a high number of numeric variables should be normalized, specially if algorithms like Neural Networks are in use (Sola and Sevilla, 1997). If the analyst is using tree based algorithms, it is not necessary to do the normalization procedure, since the algorithm looks for values for splitting points and not to the distance between values. There are several normalization techniques that range from leveling variables in approximately the same scale to aligning the probability distribution of a variable. These normalization techniques can considerably influence the final results and conclusions (Haring et al., 2007).

4. Data mining: Analytic process of extracting information from big blocks of data, exploring patterns and relations between variables. Data mining techniques involve machine learning algorithms and even mathematical relations, with the objective of creating a model that explains, in the best possible way, the distribution of the dependent variable. The data mining techniques can be used for several purposes, in this project they will be used to predict future events. During this process, several models should be developed with different bases, so that the chance of discovering the optimal solution increases.

5. Interpretation and Evaluation: After the construction of several models, the analyst needs to select the one that is most suitable for the problem in hand. In this phase, several performance indicators should be used, including the time to run the algorithm and its flexibility, depending on the objective of the project, since in the analysis phase it is very important for the analyst to know exactly what he is looking for and how to measure it.

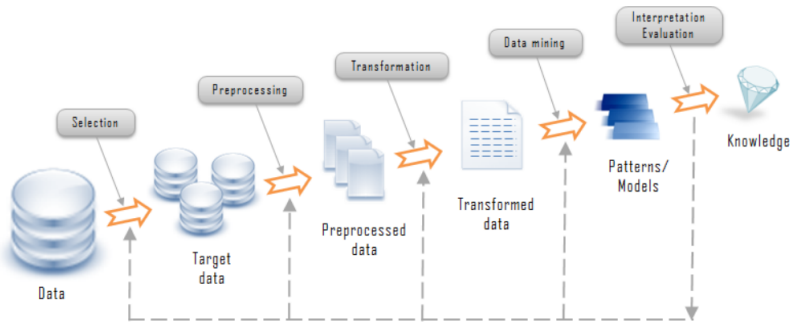


Figure 2.1: KDD process (Vera Miguéis, 2018a)

2.3 Data Mining Techniques

2.3.1 Data Subsets

For the development of prediction models it is always necessary to split the data into a train and a test set. This is obligatory because if we want to correctly test the performance of the model, we need to first feed him some data so he can learn (training set) and, secondly, test what he has learnt in unseen data (test set).

There are several ways of splitting the dataset into training and test set, but usually the training set contains the majority of the data. The two main methods are the Hold-out method and the k-fold cross validation (Kohavi et al., 1995). In the Hold-out method data is divided into two mutually exclusive sets, given a certain proportion, for example 2/3 for training and 1/3 for testing. In the k-fold cross validation method, the analyst should split the dataset into k mutually exclusive subsets, approximately of the same size, the model is then trained in k-1 of them and tested in the remaining one. This process is repeated until all the k subsets have served as the test set. The methods described above can be stratified in order to keep the same proportion of the positive and negative class as the original data, although having random samples of a big enough size should, in theory, keep the proportions.

2.3.2 Imbalanced Domains

In the vast majority of data mining problems there is a big issue of class imbalance, where one of the classes contains the big majority of the values and the other the remainder minority, and the class the analyst wants to predict is usually the one with the least observations, which can cause several problems. According to Branco et al. (2016), the two main problems of an imbalanced set are the user assigning more importance to the predictive performance of a model and the cases that are more relevant to the user being misrepresented in the training set. Exemplifying, if in 100 people, 1 has a disease and the remaining 99 are healthy, a model that predicts that everyone is healthy is 99% accurate but useless. Given this, it is usual for an analyst that uses traditional models and performance evaluations to balance the data in a way where the model that is being trained has a more even class distribution.

There are three main ways of balancing a dataset, undersampling, oversampling or a combination of both, these techniques can be done by selecting random data or using more complex methods (Branco et al., 2016). As explained by these authors, in random undersampling, a random set of the majority class examples is discarded, in order to balance the classes, this may remove useful examples and, in turn, lead to a worst performance. In random oversampling, a random set of copies of the minority class is added to the data, which can increase the likelihood of overfitting the model, which means that there is no ideal or correct way to balance the data, it depends on the problem and the algorithms in use.

There are other more complex mechanisms of balancing data, and one of the most famous is SMOTE. SMOTE is an oversampling technique where, instead of copying random samples of the

minority class, new ones are synthetically generated using the minority's class nearest neighbours (Chawla et al., 2002). The SMOTE methodology accompanies this synthetic oversampling with some degree of undersampling.

Nowadays, there are already some models that data analysts can use that know how to deal with an imbalanced class distribution on their own, in fact, balancing the dataset can even bring worse results, since the model is trained to deal with a 50/50 situation that is far from the reality.

2.3.3 Parameter tuning

The majority of the prediction algorithms have some parameters that can vary and, consequently, change the results of the model. Given that, parameter tuning is the process of changing these parameters in order to find the combination that yields the best output for the model. An ideal combination of parameters does not exist, so it is necessary to go through this process for every problem. Arcuri and Fraser (2013) showed some concerns that the user should have when tuning the parameters, since using "default" values is a reasonable and justified choice, whereas parameter tuning is a long and expensive process that may or may not pay off in the end.

In order for the model not to become overfit, it is necessary to test the different iterations in independent subsets, via a validation set. For example, the data can be split into test and training sets using the hold-out method and, inside the training set, use a k-fold cross validation to tune the parameters.

2.4 Prediction Models

Predictive modeling is a name given to a collection of mathematical techniques that have a common goal of finding a relationship between a target, response, or dependent variable and various predictor or independent variables. The goal of these models is to measure future values of those predictors and, by inserting them into the mathematical relationship, predict future values for the target (David et al., 2012).

2.4.1 Generalized Linear Model

The General Linear Model (GLM) (McCullagh and Nelder, 1989) is a generalized version of the typical linear regression, in order to encompass several others statistical models, like the logistic regression and Poisson regression. The logistic regression of GLM was used in the development of this dissertation, since it works best for classification problems, the function for the Logistic Regression is represented in Equation 2.1.

$$\pi(x) = \frac{e^{\alpha+\beta x}}{1+e^{\alpha+\beta x}} \tag{2.1}$$

The Logistic regression is used when the output is binary, there are only two possible values, and gives the probability of a certain observation belonging to either class. In order to be optimized

for a binary classification, the Logistic regression's function is a sigmoid, as exemplified in Figure 2.2.

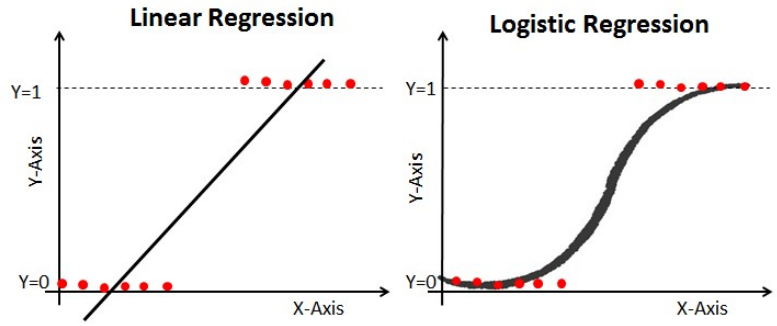


Figure 2.2: Logistic Regression Function vs. Linear Regression (Open Data Science, 2018)

2.4.2 Decision Trees

Decision trees are tree shaped diagrams mainly used to solve classification problems, they are some of the most powerful techniques in the KDD process (Bhargava et al., 2013), since they can process data with missing values and of different types (nominal, numeric and text). They are also relatively easy to comprehend for the final user.

There are several ways of creating decision trees, by splitting them using different criteria. The most commonly used factors in splitting decisions are the information gain and the gini index.

In Figure 2.3, a graphical example of a decision tree and its components can be seen. It starts with a root node containing all the data and then splits it according to a chosen variable into several nodes, the process ends in the leafs, where a certain stopping criteria decides that the tree is fully grown or if it is impossible to keep splitting.

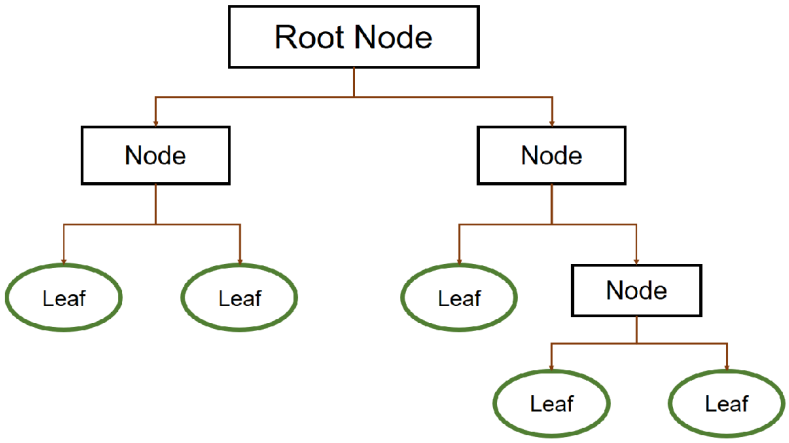


Figure 2.3: Decision Tree Example

2.4.3 Random Forests

Random forest are ensembles of decision trees, such that each tree is independent and created from a vector containing a sample of the original data (Breiman, 2001). The way an ensemble works is by evaluating an observation using several models and using the majority vote as the decision criteria. Random Forests are very popular algorithms in the data science community since they have two very important characteristics, high prediction accuracy and information on variable importance for classification (Touw et al., 2012).

2.4.4 Boosting

One technique that is frequently used in Machine Learning is Boosting and it works by turning a set of weak learners into a strong one. Some algorithms work in a similar way, like Neural Networks and Random Forests, the difference in the Boosting algorithms is that the weak models being created are added sequentially, meaning that they take into consideration the previous models before creating the new one (Natekin and Knoll, 2013). There are two main types of Boosting algorithms, Gradient and Adaptive Boosting.

When it comes to Adaptive Boosting, as Margineantu and Dietterich (1997) put it, the algorithm first creates a weak learner using equal weights for all entries. After the first learner is created, higher weights are given to the observations that were not correctly classified by the algorithm and another learner is created. This process is repeated until all the stop criteria are met. The weak learners used in the algorithm are typically decision trees, but can be anything that uses weights in the decision making.

Regarding Gradient Boosting, the process starts identically to the Adaptive one, by creating a weak learner with all the observations equally weighted (Natekin and Knoll, 2013). Following that, another weak learner is created, but, in this case, the objective of the second one is to minimize the error function of the first one, this is repeated until the stop criteria are met.

One algorithm that has appeared recently and has proven to be one of the strongest, is an optimization of the gradient boost one, the extreme gradient boost (XGBoost). XGBoost applies some techniques to reduce the overfitting of the models, like some penalization to the trees and some trimming of the leaf nodes. It also adds an extra randomization parameter to the generation of the trees, that helps reduce their correlation and, in turn, improve the results. Although XGBoost is generally a faster learner than the normal Gradient Boost, it is not guaranteed to always outperform it, the regular Gradient Boost is a very solid algorithm with a wide range of applications.

2.4.5 Neural Networks

Artificial Neural Networks (NN) are mathematical models trying to emulate biological neural systems (Singh and Chauhan, 2009). They are not actual prediction models, but a system that allows several algorithms to work together and learn to solve the proposed problems. One of the most common uses of neural networks is in image recognition, but it can be applied to any Machine Learning problem.

NN are comprised of an input layer, an output layer and at least one hidden layer. Hidden layers are formed by one or more nodes and each node is, simply put, an equation, usually a sigmoid function. The data enters via the input layer and goes, with a certain weight, to the hidden layers, where it will decide which output to give. As the models learns, the different weights between layers are updated. A NN with two hidden layers can be seen in Figure 2.4.

Although NN are the most powerful algorithm, they are also the most complex to explain, especially when it enters a deep learning state (several hidden layers) and this can potentially be problematic. They provide better results with data that is normalized and balanced.

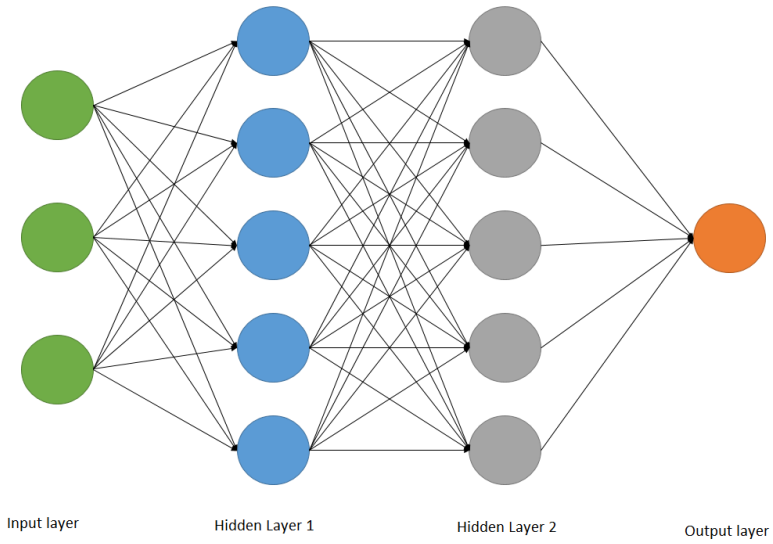


Figure 2.4: Neural Network

2.5 Performance Indicators

Performance Indicators are uniform for all the models described above which makes it easy to compare several models together. Depending on the context of the problem, some indicators can be more useful than others. The most commonly used indicators will be described in the following sections.

2.5.1 Accuracy

Accuracy is one of the most commonly used indicators to evaluate the performance of a model. The accuracy measures the percentage of times that the model correctly predicts the dependent variable. The formula to calculate the accuracy can be seen in Equation 2.2.

This model is not always the best for practical problems, since it only works well in balanced datasets, if the classes are imbalanced this indicator gives little to no insight about the quality of the model (Liu et al., 2015). In fact, in certain cases low accuracy may be preferred over a higher one. For example, if in 100 patients, one has an infectious disease and the rest are healthy and

we are trying to predict which one it is, if the model predicts everyone to be healthy, it has a 99% accuracy but does not give useful information. If it predicts that 10 patients are sick and 90 are healthy, the accuracy decreases to 91% but allows the search to be reduced from 100 to 10 patients.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{2.2}$$

2.5.2 Specificity, Recall (or Sensitivity) and Precision

Specificity is the proportion of True Negatives (TN) in relation to the total number of negatives, which means that the least False Positives (FP) you have, the better the value of this indicator. This is always relevant when we want to know how many negative values our models correctly predicted. The formula for specificity can be seen in Equation 2.3.

$$Specificity = \frac{TN}{FP+TN} \tag{2.3}$$

Recall is similar to specificity but for True Positives (TP) in relation to the total number of positives, meaning that to have a good value for Recall, the least amount of False Negatives (FN) is needed, as can be seen in equation 2.4. This indicator is especially relevant when the detection of the majority of the positive class of the problem is necessary.

$$Recall = \frac{TP}{TP+FN} \tag{2.4}$$

Precision is the proportion of positive answers that are actually relevant for the problem in hand. Precision is calculated following Equation 2.5, this indicator is relevant if the analyst wants to penalize the FP of a specific model.

$$Precision = \frac{TP}{TP+FP} \tag{2.5}$$

Neither of these three indicators is recommended for main performance indicator of a problem, especially if the classes are imbalanced, since they only evaluate half of the confusion matrix, but they are all very useful when combined with other indicators since their comprehension is simple and valuable.

2.5.3 F-score

Lately there has been some discussion about the utility of the more traditional performance metrics, like the ones described in the previous sections, since they all present a trade-off and make it hard to analyze more than one parameter at once. As Liu et al. (2015) explains, precision and recall are negatively correlated and influenced by what we want to predict, if we just use one of them, they have little to no meaning. Given that, performance metrics like F-score appear, a measure calculated as the harmonic mean between Recall and Precision, represented in Equation 2.6, with a variable parameter β that can take several values, depending on the weight the analyst

wants to give to Precision and Recall. Since this equation utilizes more than one of the previously described parameters, it is more adequate to use in imbalanced datasets.

$$Fscore = (1+\beta^2)\frac{precision * recall}{\beta^2 * precision + recall} \tag{2.6}$$

2.5.4 Matthews correlation coefficient

Matthews Correlation Coefficient (MCC) is used mainly in binary problems with imbalanced datasets. As stated by Liu et al. (2015), it returns a value between -1 and 1, where 1 means that the model correctly predicted 100% of the observations, 0 represents an efficiency similar to a random guess and -1 signifies that all observations were incorrectly predicted. MCC uses all of the entries in the confusion matrix in its Equation (Equation 2.7). Although it is impossible to describe the entire confusion matrix in just one value, the MCC is the indicator that most closely resembles that.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(FP+FN)(TP+FP)(TN+FP)(TN+FN)}} \tag{2.7}$$

2.5.5 ROC curve

Receiver Operating Characteristic (ROC) curve represents the trade-off between TP and FP for every value of threshold, an example of this curve can be seen in Figure 2.5. According to Ling et al. (2003), it compares the performance of the models across the entire range of class distributions and error costs. The ideal situation is to find a threshold value where the point in the graph is (0,1), 100% TP and 0 FP, this is almost impossible, so the value that comes closer to that is the ideal threshold value for the problem.

Using the ROC curve, the analyst can also calculate another metric, the Area Under Curve (AUC) which is the value of the area below the ROC curve, this value can be compared to the accuracy but is normally consider a better performance metric. Ling et al. (2003) did a study comparing accuracy to AUC and found out, with statistical significance, that AUC is a better measure for evaluating learning algorithms than accuracy.

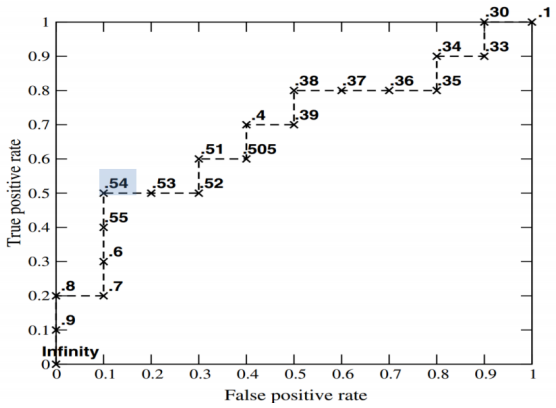


Figure 2.5: ROC curve example (Vera Miguéis, 2018b)

Based on what was mentioned in the current and previous sections, the goal of this project is to use data mining techniques in order to predict customer's Service Requests in a telecommunications company. During this dissertation, several algorithms and data balancing methods will be used and tested with the intent of finding which provides better results, as well as an in depth discussion about what metrics should be used to evaluate the performance of certain models.

Chapter 3

Problem Description

The current chapter presents a detailed description of how the problem is currently handled, an extensive analysis of what characterizes the billing SR and a description of possible improvement points.

3.1 Customer Satisfaction

One widely used metric for customer satisfaction is the Net Promoter Score (NPS) and it is currently used in the company where this project was developed. It works by asking costumers "How likely are you to recommend the company to a friend/family?", from a scale from 1 to 10. If the respondent gives a mark of 9 or 10, then he is a promoter, 7 or 8 he is passive and those who give a grade of 6 or lower are considered detractors. NPS is calculated by subtracting the percentage of detractors to the percentage of promoters and thus ranges between -100% and 100%.

According to Azzarello and Kovac (2011), the company's promoters and NPS measure are crucial for its success, companies with a higher NPS have customers who buy more, stay longer with the company and are more likely to bring other people to the company. Promoters are also known for giving useful feedback and help improve the business. It was also found that promoters generate 80 % more lifetime revenue than the passives and 250 % more than the detractors.

Given this, it is expected that a company should strive for the increase of its NPS. The partner company is currently aiming to improve this indicator as much as possible, starting by an initial analysis of what the customers perceive as essential for a telecommunication company. Figure 3.1 shows the results of a study inquiring the clients about that (multiple choices could be selected).

One of the biggest goals for the partner company is to solve the "Correct/Clear Bills" problem and, as a consequence, help in solving the "Price defined is the price paid" as well. The project developed during this thesis is a tool to help solve this situation with the intent of increasing the overall NPS.

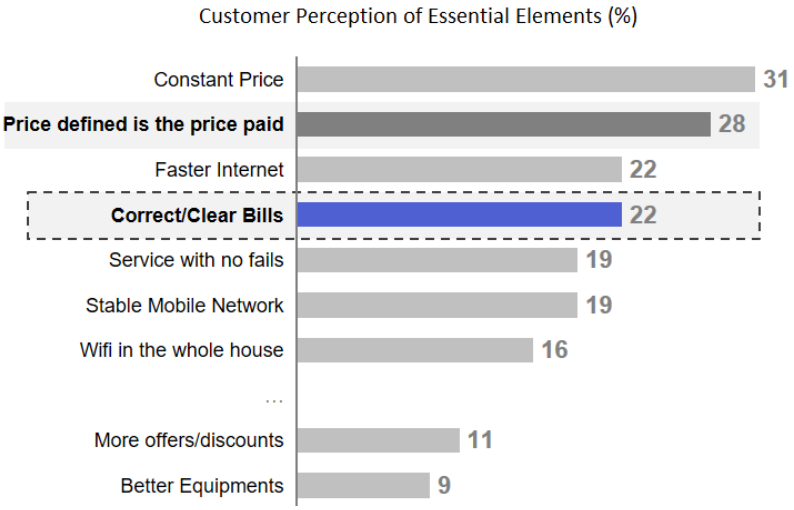


Figure 3.1: Essential Elements rating (Partner company's internal documents, 2019b)

3.2 Current Situation

Nowadays, SR are received mainly by the customer service or directly by the retail stores, although other departments can receive service requests as well. Figure 3.2 illustrates the number of SR received by each department: CS (Customer Service), Retail, Commercial, FD (Financial Department) and Others (internal departments). These doubts are registered and ideally solved in the first contact but, if they are to complex, they are transferred to other departments more capable of handling them. The duration of the problem is registered, if it was handled at first contact, the details and characteristics of the client, if it is a recurring doubt and the client's evaluation of the customer service.

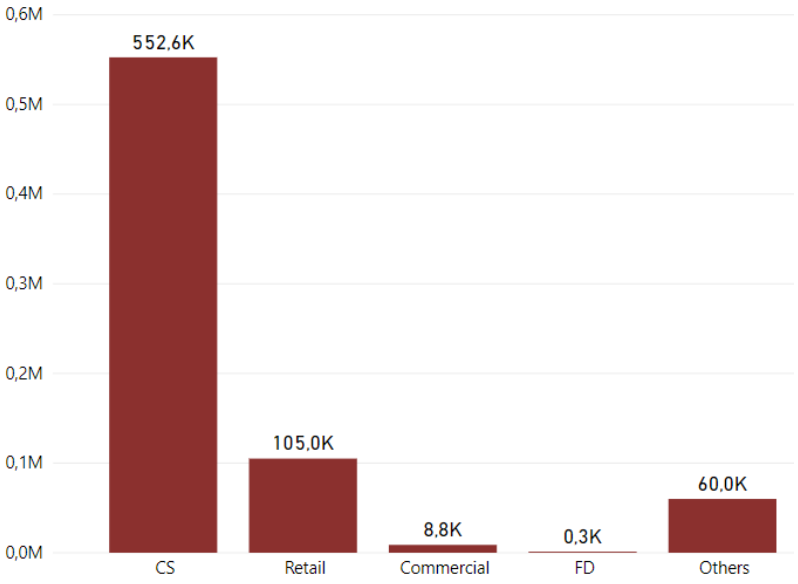


Figure 3.2: Service Requests Received per Department

Of all the yearly service requests the partner company gets, around one million are about billing. When a SR is received, it is typified by the employee who initially handles it, there are 98 main billing typifications that can be subdivided into even more subcategories. Of this 98, the majority represents a very small percentage of the total number, since there are three main categories that cover more than 50% of all SR, as can be seen in Figure 3.3.

The biggest category, Billing values, is responsible for more than 30% of all cases since it is the more generic one, representing every SR where the client does not agree with the bill value or does not understand it. More than half of these situations are misunderstandings where a simple explanation of the value is enough to solve the situation. The second biggest topic to be handled is the Promised Commercial Conditions, covering 12,4% of all billing SR. These situations arise when a client is promised a free equipment, channel or credit and does not see it reflected on the bill, this can happen due to an actual error or simply because of the billing procedure, where the discounts sometimes are only reflected on following bills. The third most relevant topic, covering around 9% of the situations relates to mobile voice consumption. The clients contact the partner company due to an extra value on their mobile's bill that they do not think should be there but that more than 90% of the time ends up being a correct debit. The remaining typologies are very diverse, some cover clients who asked for the service to be shut down but were still billed, others cover bills with duplicate values, services that were not requested but were billed, there are also typologies for clients who were charged a full month but the service had been down some days and many others.

It should be noted that there are definitely some clients who have doubts or complaints regarding their bills but do not contact the partner company's services. These clients are potential false positives, meaning that in the context of this problem, the false positive observations are dubious and should not be considered as complete misclassifications.

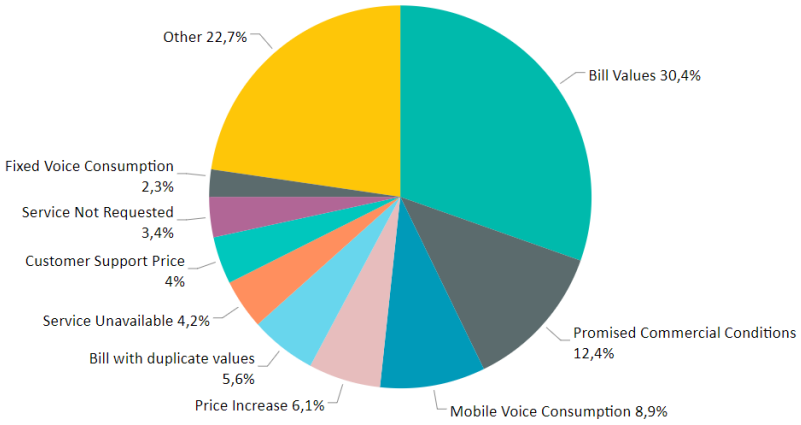


Figure 3.3: Service Requests Typifications

By analyzing a full year of SR (October 2017 to September 2018), it is clear that the partner company has a huge volume to process, the number of service requests (SR) related to billing, for every hour, for this period, can be seen in Figure 3.4 . There is no clear peak hour for SR, it is a

generalized problem and thus, the solution should pass for an overall decrease of the volume and not only by a reallocation of the resources.

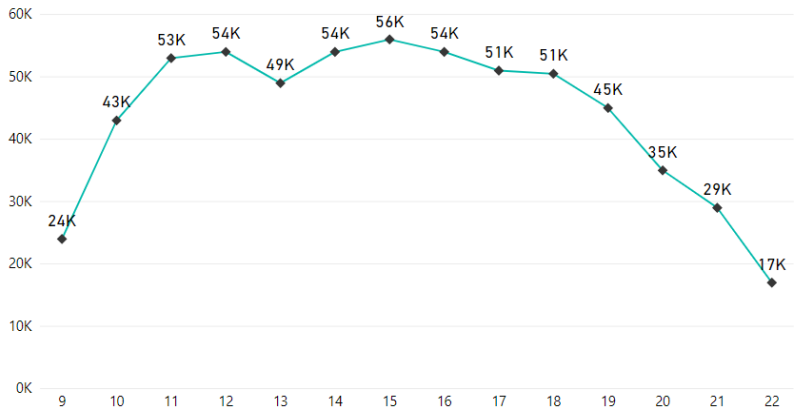


Figure 3.4: Service Requests each hour

An initial data analysis reveals some patterns, for example, clients are more likely to need a service request immediately after starting the service or changing their bundle, there are no clear peaks during the month and there are clients more communicative and others more silent. The next section presents a detailed view of this analysis.

3.3 Analysis of Customer's Service Requests

The analysis of the frequency of the calls revealed that although there are some periods with more SR than others, there are no extreme peaks during the month. The days were the volume of service requests is higher is in the beginning and in the end of each the month, between day six and seventeen of any month, the volume is lower. Figure 3.5 shows the number of SR received each day for an entire year (October 2017 to September 2018). The reason is that there are various billing cycles throughout the month and people receive their bills somewhat evenly.

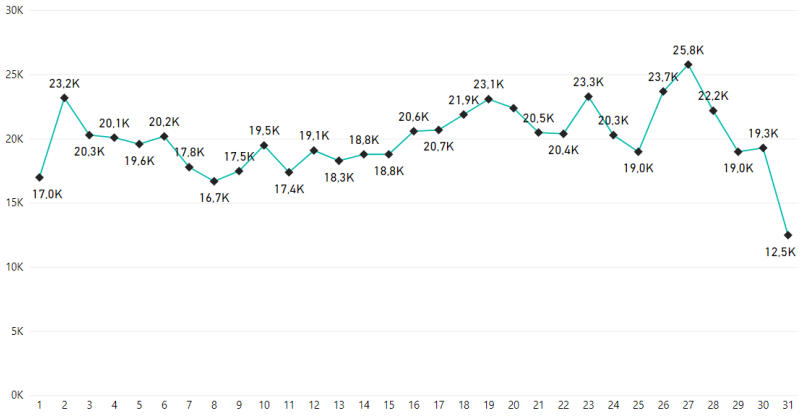


Figure 3.5: Service Requests each day

As mentioned in the previous section, the majority of the service requests enter through the customer service, in fact, more than 75% off all service requests come via call to the customer service department, the retail stores receive around 15% of the doubts and the remainder are distributed throughout several departments.

The city who receives the biggest amount of service requests, as expected due to its population size, is Lisbon, with 30%, followed by Porto, with around 20%. Setúbal, Braga and Aveiro come next, with a combined total of 25% of the billing SR. Moreover, it should also be noted that around two thirds of the service requests are handled within the first contact with the partner company, the rest are more complicated problems that the employees answering the client can not fix on their own.

The partner company's clients do not need to provide their age when they subscribe to a service, so the values representing their ages are estimated using the customers' NIF. As such, the information of the customer's age will not be used in the algorithm, since it is somewhat inaccurate and impossible to validate.

As mentioned above, there are some factors that clearly increase the chance of a client asking for a service request, like recently joining or changing bundle. In fact, each month, the percentage of clients that have a question is about 3%, when selecting the clients who have just joined, this percentage triples, to around 9%. The clients who have changed their service bundles also have an even higher percentage of asking for a service request, more than 10,5%.

Lastly, there seems to be a clear distinction between clients who call and clients who do not, meaning that a customer that has called in the previous month has a bigger chance of calling again, in fact, five times bigger, almost reaching a 15% chance. The clients which are more silent and have not called in previous months, have a lower than average chance of calling, around 2,5%. In Figures 3.6 and 3.7, a more visual representation of these relations can be seen.

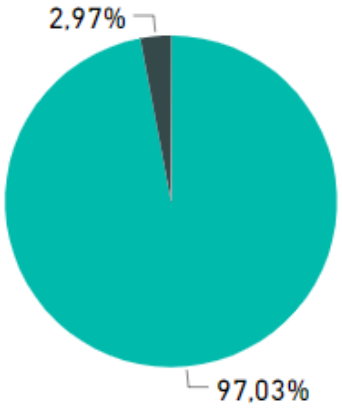
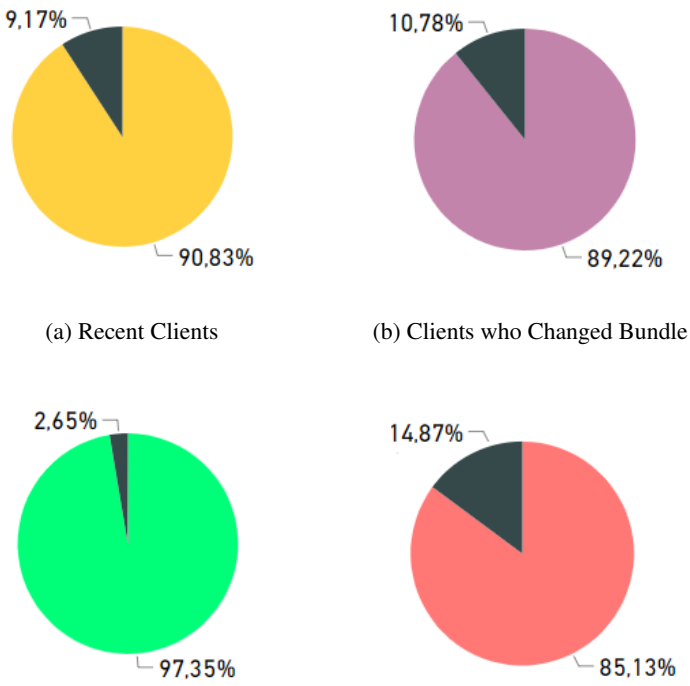


Figure 3.6: All Service Requests



(a) Recent Clients (b) Clients who Changed Bundle

(c) No Recent Service Requests (d) Recurring Service Requests

Figure 3.7: Percentage of Service Requests in the Client Base

3.4 Improvement Points

The analysis enabled the identification of several opportunities for improvement that can potentially increase the efficiency of the service request handling process:

1. Reducing the number of billing SR presented by clients;
2. Giving information to the client in a clearer way;
3. Distributing customer service resources more evenly.

The project's main objective is to create a tool to help to solve the first problem, by creating an algorithm that predicts clients' emerging doubts, allowing the company to solve them before the customer complains. It is expected that it is possible to identify the clients more likely to call and solve their issue on the partner company's terms, hence reducing the number of service requests received every day.

The second improvement point is not a main objective of this analysis, but it can be solved as a consequence of the predictive modeling. By analyzing what variables are more important for the model, it is possible to deduce what aspects of the bill create more confusion in the client base. Instead of a complete redesign of the billing procedure, with a better understanding of what confuses the clients, the topics most likely to generate services requests will be identified and help

in defining which parts of the bill need to be clearer. For example, the analytic analysis showed, as mentioned in previous sections, that a change of service bundle can create doubts, in those cases, the bill should be accompanied with an explanation of the changes.

The third topic involves the more even distribution of the customer service departments' resources, namely people and money. As mentioned in Section 3.2, the number of billing SR received each hour is similar and such, it would be hard to redistribute the resources but, although this is the case for billing SR, the partner company receives many other SR regarding different topics. Only after collecting that information would it be possible to know if there are more problematic hours and, if it is possible to redistribute the resources in a way where they are even through out the day.

To summarize, it is clear that the problem exists and that the billing SR are not random, there are some patterns when it comes to cause and customer behaviour. It can also be concluded that the customer perceives the situation as problematic and that solving it would definitely increase the company's overall NPS.

Chapter 4

Project Development

After the problem has been clearly defined and the objectives stated, the following step is the construction of the model. During this chapter, an extensive analysis of all the steps involved in the creation and optimization of a prediction model will be explained. There will also be a description of the methodology used in the project as well as the tools that were needed for the procedure.

4.1 Methodology

The typical approach for any data mining project is to use the KDD methodology, described in Chapter 2, although this methodology is the standard, it is theoretical. When analyzing a real business problem, it is necessary to always keep in mind the goal of the project and the idea behind its implementation. The Cross-industry Standard Process for data mining (CRISP-DM) is an adaptation of the KDD process to a more business oriented approach. In Figure 4.1, a detailed diagram of the process is shown. With the initial and reoccurring phase of business understanding, and the final phase of deployment, CRISP-DM takes the base of the KDD procedure and enhances it to a more practical approach, it can be seen as an implementation of the KDD process (Azevedo and Santos, 2008).

The first step of this methodology is understanding the business, it is crucial in any corporate level problem to know why the project is initiated and what is its goal. In this project, the business understanding phase is represented in Chapter 3, with a description of the initial situation and the objectives of the project. The following Data Understanding, Data Preparation and Modeling phases are explained in depth in the current Chapter 4. Finally, in Chapter 5, the Evaluation phase is fully analyzed.

The project was developed using the R Studio software, utilizing several resource packages, namely H2O (LeDell et al., 2019), XGBoost (Chen et al., 2019), caret (from Jed Wing et al., 2018) and sqldf (Grothendieck, 2017). The visualizations were done with Power BI and Microsoft Excel.

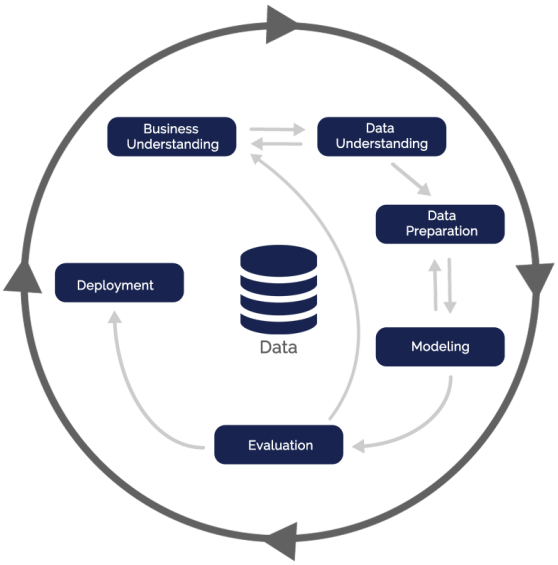


Figure 4.1: CRISP-DM diagram (Otaris, 2018)

4.2 Data selection and Preprocessing

After the business understanding phase of CRISP-DM, the first step to any predictive modeling project, as mentioned in Section 2.2, is the selection of a suitable dataset for the analysis. In the CRISP-DM methodology, this phase is split into Data Understanding and Data Preparation. It is then necessary to think about what variables, or features, would most benefit the problem in hand. Considering this, every information possible about the clients' account would probably benefit the model, specially if it involves portfolio changes that affect the bill. Other than that, it would also be interesting to understand if personal characteristics, like gender or age have an impact on the prediction.

The partner company is very conservative in what it requires from its client base and hence, personal characteristics, as the ones mentioned above are not readily available. Although they can be estimated, the values will never be 100% reliable. As far as the account characteristics go, everything is available, but only what the client has or had in every moment, to have the information about what changed from one month to the other, the analyst needs to do a comparison. Since having a variable representing these changes would probably be of value to the the model, some feature engineering will need to be employed, a step which will be described in a latter section. Lastly, it is crucial to know the values of our dependent variable, for that, the information regarding the billing SR needs to be obtained. Since that information is kept already by typifications, it can easily be filtered by Billing.

To construct the dataset, the data from the clients' sheet was collected, this sheet contains a lot of useful information, like what bundle does the client subscribe to, the speed of his internet, the cost of his bundle, what premium services does he subscribe to and others, but it also has some data that can and should be disregarded, as it would not add any value to the model and would only slow him down. Regarding that, since the data is already clean and organized, with no noise or

missing values, the preprocessing stage will focus mainly on removing the useless columns, like several ratios, included in the client's sheet, calculated for other purposes and that are of no value to the current study.

4.3 Performance Metrics

There is no definite answer for what metrics should be used to analyze a problem, as mentioned in section 2.5, there are several and all give different insights, it is up to the analyst to determine exactly what he wants to measure and which metrics provide that information. In this project, the MCC was used as the main performance metric, since it can evaluate the entirety of the confusion matrix and is often seen as its representation in one value. MCC is also known for being a good way to measure performance in unbalanced domains. The utilization of this metric was also convenient since it is the main indicator used by the Data Science team at the partner company, making it more suitable to present and compare results.

One other metric used for analyzing results was the Lift. Lift is not an academic measure but it is often used in corporations, since it is more directed to the implementation of a solution and also a clearer way to present the results to individuals not familiar with predictive modeling, as is often the case at the partner company, the formula for lift can be found in Equation 4.1. This measure works on the premise that the solution found can not be applied to the entire population, since there are limitations in terms of technology and resources. The analyst selects a K percentage of the population, usually 10% and tries to include the majority of the positive class in that percentage. Lift is calculated by determining how much more percentage of the positive class is in the first k % than in the overall population. For example, if the baseline is 7% and the Lift for k=10 is 8, it means that in the top 10% of the predictions, we can have a precision of 7*8= 56%. In Figure 4.2 there is a visual representation of the Lift, where the analyst, in his top 10% of predictions has a density of 29% positives, compared to the baseline of 8%, meaning that his Lift is 3,625.

$$Lift = \frac{precision\ top\ _{k\%}}{\%\ positive\ cases},\ k\ \in\ [1,100] \tag{4.1}$$

4.4 Predictive Modeling

In order to have a good sense of the data and what models fit it better, it is crucial to select models with different bases, in this case, linear, tree based and neural networks. A generalized linear model (GLM) was employed, using a Logistic regression to try and fit the model, as well as a Random Forest (RF), a Gradient Boosting (GB), an Extreme Gradient Boost (XGBoost) and a Neural Network (NN). As mentioned in Section 2.4.3, XGBoost is an optimized version of the GB algorithm, but both of them will be used for testing in this project since GB is also a solid algorithm. It is expected that XGBoost will outperform GB, but it will be a way of comparing these two methods and testing this hypothesis in a practical scenario.

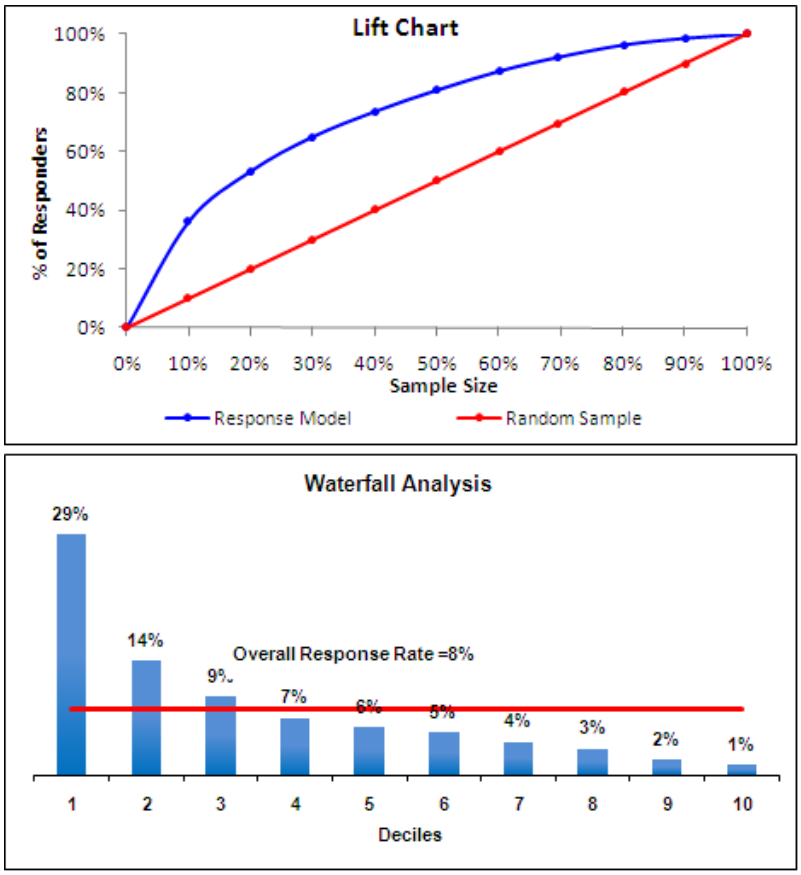
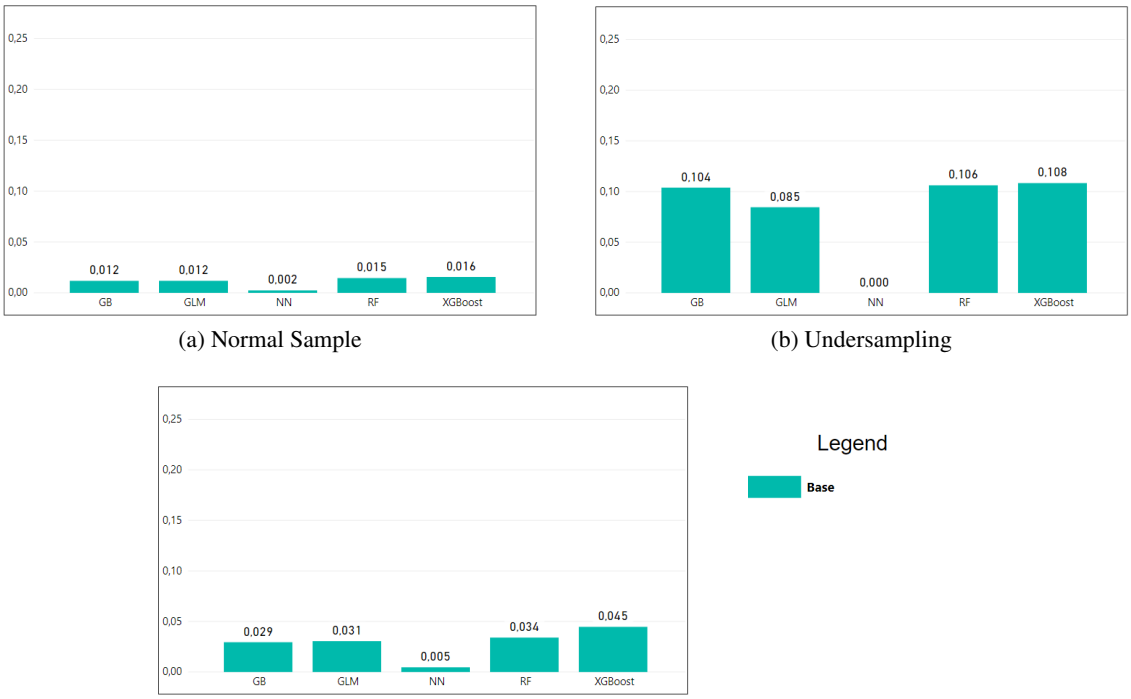


Figure 4.2: Example of a Lift Chart (StackExchange, 2011)

It is then necessary to choose what performance indicators to use in order to compare the several algorithms, as mentioned in the previous section, MCC and Lift are overall good options, other than that, the more basic indicators, like accuracy, precision, recall and f-score were also calculated in order to have a constant sense of the problem's ratios. The dataset was constructed using information from the months of July and August of 2018, since this was the data available for modeling, and was split by month, meaning that July 2018 was used as the training set and August 2018 as the test set.

4.4.1 Baseline Models

The first models developed were using the basic data, in order to have a benchmark to work from. The five algorithms, listed in Section 4.4, were used using default parameters and different balancing techniques, original sample, undersampling until a 50/50 distribution and oversampling by tripling the positive samples, the prediction threshold was set at 0,5. The results obtained with the first models are represented in Figure 4.3 (Detailed Results in Appendix A Figure A.1). It should be noted that when the NN were constructed, the numeric variables of the data were normalized in order to fit in a scale of 0 to 1, the normalization was applied to the training set and then the formula was used for the test set.



(a) Normal Sample

(b) Undersampling

(c) Oversampling

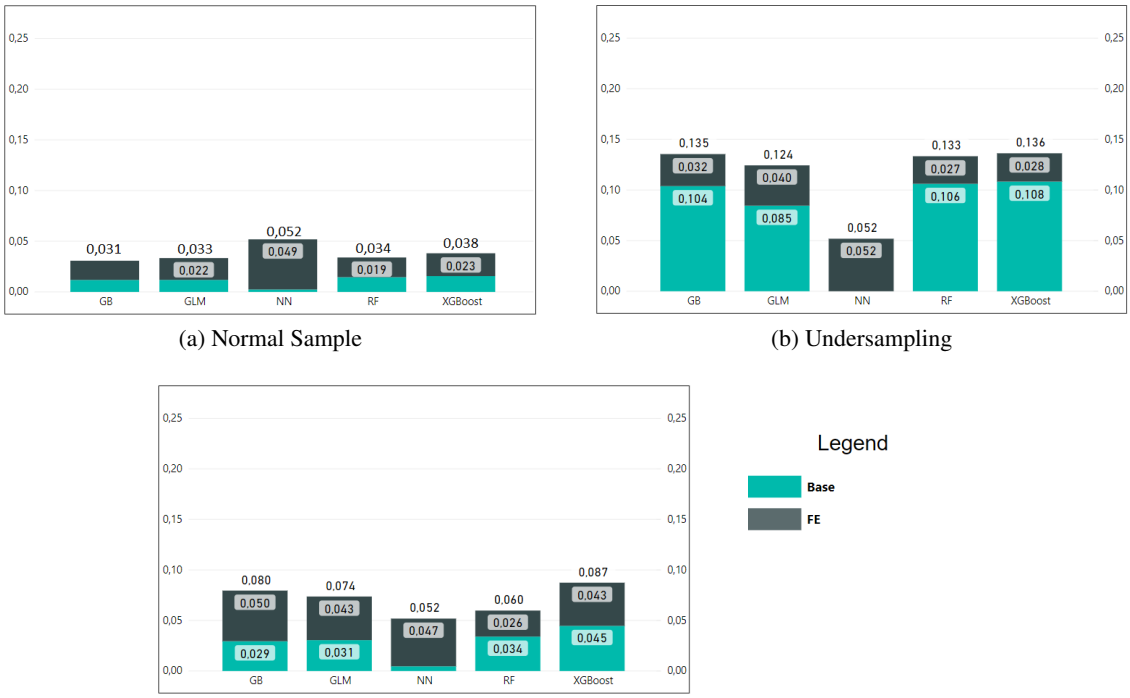
Figure 4.3: Base Results

Looking at the results, they are, as expected, not very good, the models that received no balancing technique had a MCC close to 0, resembling a random guess. With oversampling, the results were marginally better but, the undersampling technique yielded the best results, with an MCC of around 0,1 for all models except the Neural Networks, where the model could not explain anything so it just predicted all positive values, due to the lack of meaningful variables and a short run time (default parameters). The results with no balancing technique were very low since almost no positives were predicted, which can be a consequence of the threshold being too high.

4.4.2 Early Feature Engineering

The first step to improve the baseline results is to do some feature engineering (FE), it is important to understand what causes clients to submit service requests, in order to create relevant variables. As mentioned in Chapter 3, new clients and clients who changed bundles have a higher chance of contacting the partner company. Other than that, clients who contacted in the previous month are also more likely to do it in the current one and thus, variables containing that information were added into the dataset. To conclude the initial feature engineering, a variable containing the money spent on extras that month was also included, since some clients might subscribe to premium channels or pass their mobile data limit without noticing. Adding those variables yielded the results in Figure 4.4 (Detailed Results in Appendix A Figure A.2).

The results obtained are overall better, maintaining the patterns that were observed in the baseline instance. Using the default parameters for the algorithms, Gradient Boost and Extreme



(a) Normal Sample

(b) Undersampling

(c) Oversampling

Figure 4.4: Results after Feature Engineering

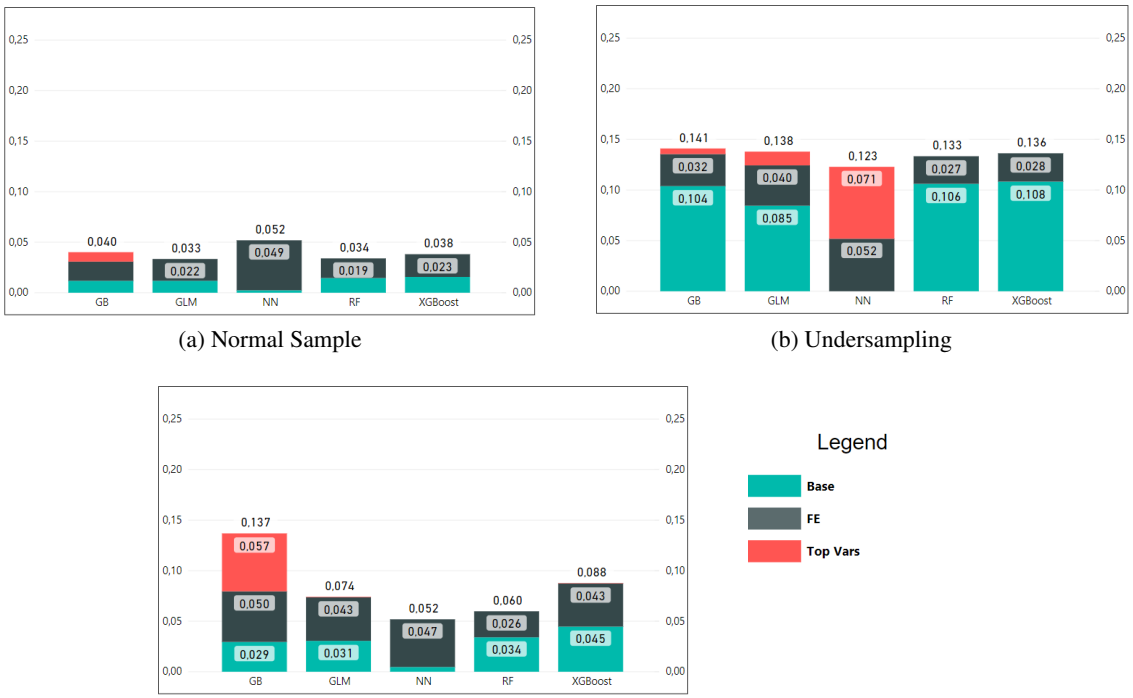
Gradient Boost yielded the best results.

4.4.3 Feature Importance

At this point, it is important to understand if all the variables being used are important to the model, and such, it is crucial to rank them for their contribution to the results. Since the models require a long time to run, reducing the complexity of the dataset by removing features that do not have a big contribution to the solution can be a big help. With that, for each of the five models, the ten variables with the least importance were selected and compared with each other, the ones that were common for all five algorithms were removed from the analysis. A total of eight variables were removed. The results obtained after removing these variables are represented in Figure 4.5 (Detailed Results in Appendix A Figure A.3).

Removing these eight variables helped speeding up the algorithm, reducing the average run time by around 5%. The results obtained were not better than the previous ones, in a meaningful way, meaning that the removed variables were not disrupting the algorithm, but removing them also did not penalize the models. There were only a couple of algorithms that really benefited from this, the undersampled NN and the oversampled GB. In conclusion, the eight variables removed had no significant impact in terms of loss of performance and thus should be excluded from further analysis.

By analyzing the results, it is possible to conclude, once again, that the threshold might be too strict, since the big majority of the predictions for the normal sample are of the negative



(a) Normal Sample

(b) Undersampling

(c) Oversampling

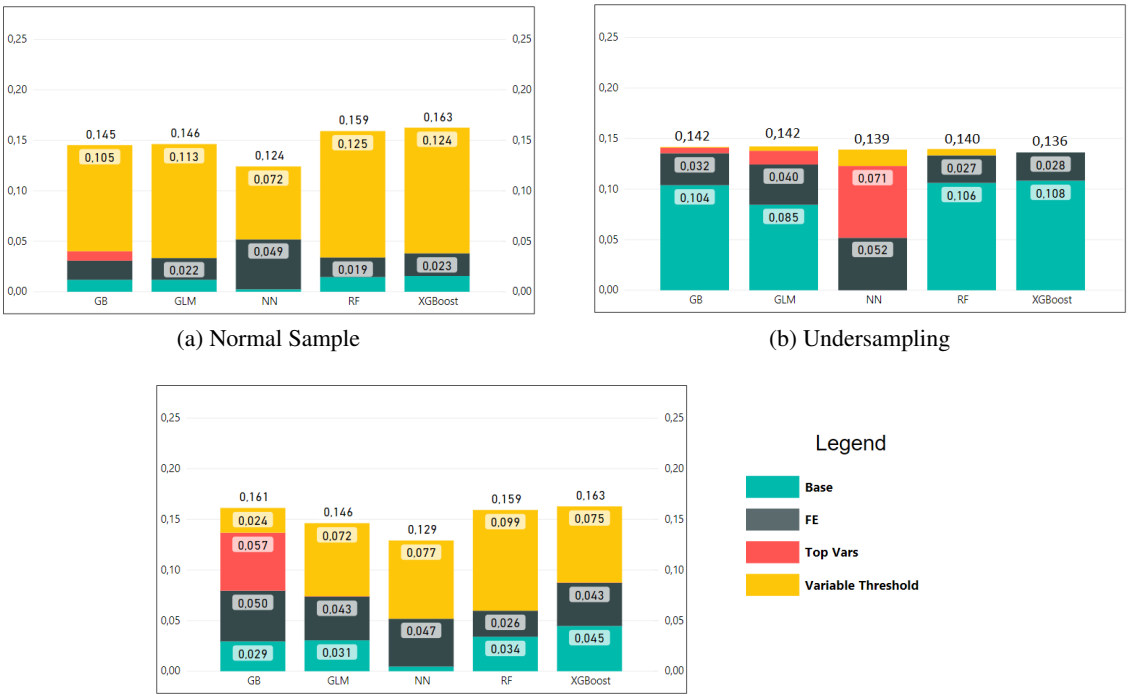
Figure 4.5: Results with Top Variables

class. As such, another round of test was made, with the threshold being optimized to a value that maximizes the MCC indicator in the training set and then applied to the class allocation in the test set. The results, shown in Figure 4.6 (Detailed Results in Appendix A Figure A.4), prove that with a variable threshold they can improve substantially. The Gradient boost algorithms (Normal and Extreme) continue to be the ones that output the best values, but the undersampling technique is no longer the best performing, in fact, it was surpassed by both the oversampling and the regular one. Further analysis should take this into consideration and always use a threshold optimized for MCC when allocation the results.

4.4.4 In Depth Feature Engineering

In order to further improve the results, it is necessary to add more relevant features to the model, thus, variables similar to the ones created above were made, but instead of comparing only with the previous month, they will do it with the last three. Other than that, the number of SR made by a client in the previous three months will also be added to the model, since the analysis made in Chapter 3 revealed that some clients do indeed contact the partner company more frequently than others. Lastly, the average payment for the last three months as well as the difference between it and the current payment will be new features in the model. The results for this new step of Feature Engineering are represented in Figure 4.7 (Detailed Results in Appendix A Figure A.5).

The new variables added were successful in improving the overall results. The undersampling technique continued to prove it is the weakest, and oversampling yielded results very similar to



(a) Normal Sample

(b) Undersampling

(c) Oversampling

Figure 4.6: Results with a Variable Threshold

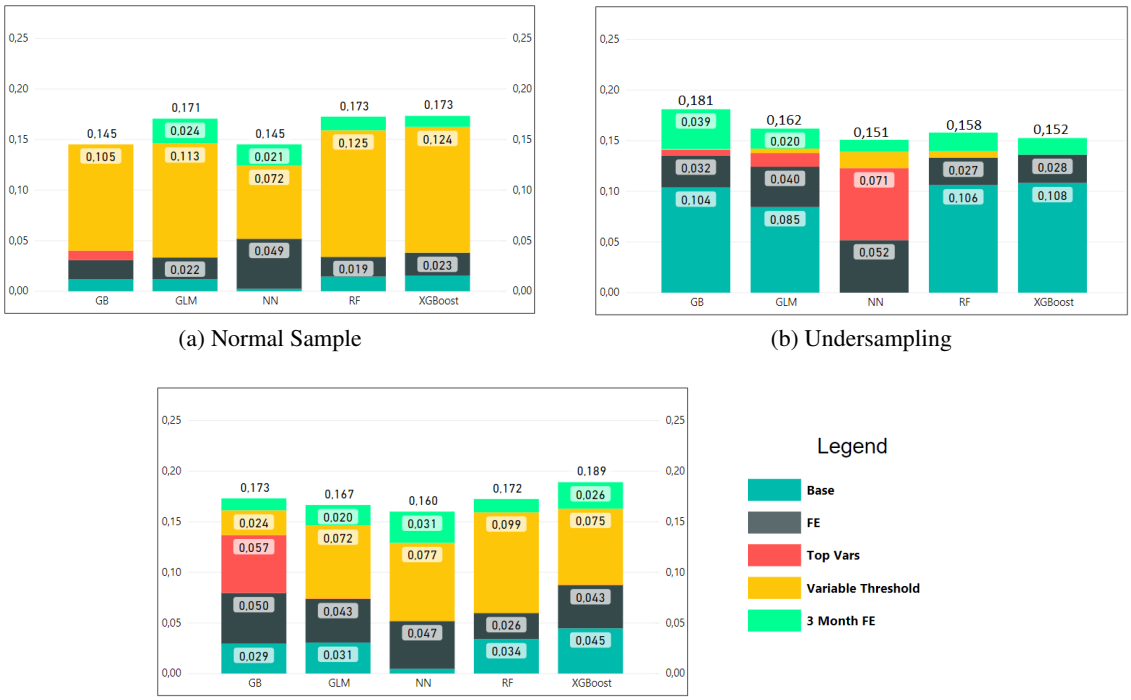
the ones without sampling techniques. Considering that, for further analysis, in order to speed up the process, no sampling technique will be used, only the real class distribution. In the final step of the analysis, when all the feature engineering is complete and the parameters ready to be tuned, the final model will again be tested with class balancing techniques.

To decide what further features to add to the model, a detailed analysis of the predictions should be done, to understand what is missing. In order to do this, a list of the top 20 worst predictions was made. This list was made by ordering the output by increasing probability of being from the positive class and collecting the first 20 observations that were, in reality, positive. In other words, the 20 observations that should belong to the positive class, but the model predicted with the most certainty that they would be negative.

By observing these 20 entries' specific typification, it is easier to understand why they were incorrectly assigned (Figure 4.8).

The type of SR that is most represented in the 20 worst predictions is Bill Values, the typology that is also the most represented in the overall population of SR, the second most represented class is Promised Commercial Conditions, which is natural since in the current variables there is none about campaigns or discounts. The Bill with duplicate values typology is also represented in this analysis, this type of SR is a symptom of billing errors and their prediction is not a direct objective of this project.

The remaining categories involve Churn Requests, Service Failures and also Maintenance



(a) Normal Sample

(b) Undersampling

(c) Oversampling

Figure 4.7: Results with a 3 Month Comparison

Costs. Currently the model has no information about these problems and thus, adding that information will help it reach better conclusions.

Data about other types of service requests, involving the problems mentioned above, was collected and inserted in the algorithm. Ten new variables were created, 2 about Service Failures, another 2 about churn requests, 2 more about maintenance interventions and the last 4 about commercial conditions. These variables were added separately, to monitor their impact individually.

All of the variable showed that they individually could enhance the results obtained except for the ones related to maintenance interventions, that caused no improvement. A detailed view of all the results can be seen in Appendix B.

As expected, since all the variables that were positive for the model were added, the algorithm was able to identify more clients that would present SR, improving the results, shown in Figure 4.9 (Detailed Results in Appendix B Figure B.5). The results when using the RF algorithm are lower than expected since it is, in theory, one of the best algorithms for predictive modeling problems. This can be due to the default parameters in use not being the optimal for the problem in hand and thus, it is expected that this algorithm will be one of the ones with the biggest improvement after the hyperparameter tuning phase.

4.4.5 Hyperparameter Tuning

The last phase of the modeling process involves the optimization of the algorithms' hyperparameters. For each of the five algorithms in use, several parameters were changed in order to try and

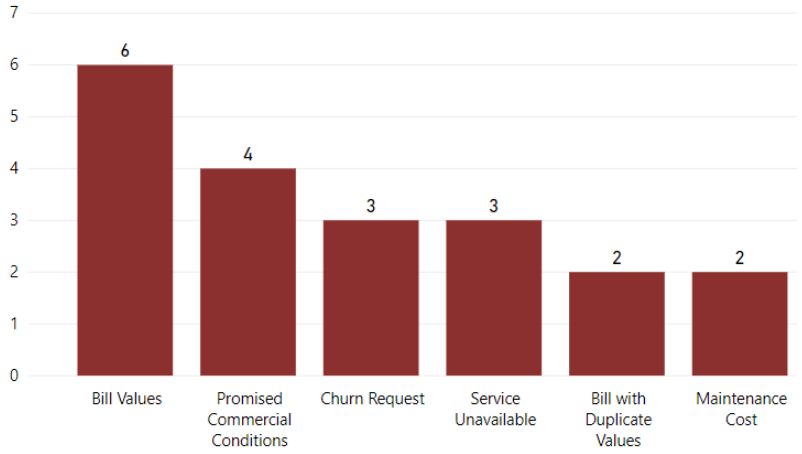


Figure 4.8: Typologies of the 20 worst predictions.

improve the results, as well as to discover the algorithm that yields the best outcome. This is necessary since the default values for each parameter can be far from optimal, either not letting it learn as efficiently as it should or even allowing the model to overfit. Given that, it is important to test different combinations of these hyperparameters to ensure that an optimum is found.

When developing any Machine Learning model, it is crucial to split the data into training and test set, where the test set needs to be excluded from the construction of the model and only utilized to obtain the final results. Since the Hyperparameter tuning requires several models to be built and compared in terms of performance, it is impossible to do so with just a training and test set, a validation set also needs to be utilized. A validation set can be created by removing part of the training set and using it for validation.

In order to obtain a more reliable estimate of the results for each set of parameters the k-fold cross validation technique was employed. The k-fold cross validation has been extensively tested and is considered one of the best techniques for finding the optimal models for certain problems (Jung and Hu, 2015). In this technique, the dataset is split into k-folds, where one is used for testing and the remainder for training. The process is then repeated until all folds have served as the test set, creating k results that are then averaged (Figure 4.10).

When applying the k-fold cross validation technique to Hyperparameter tuning, the data is split into test and training set and then the training set is split again into k folds, where one is used as a validation set and the rest as training, the process is then developed as explained in the previous paragraph. For this project, five folds were used for this technique, since less than that could not be significant enough and more would require an extremely high computing time, due to the size of the dataset and limited resources.

4.4.5.1 Grid Search

A grid search works by sequentially going through all the possible combinations of hyperparameters selected by the analyst, creating a model with them and testing in the validation frame. After

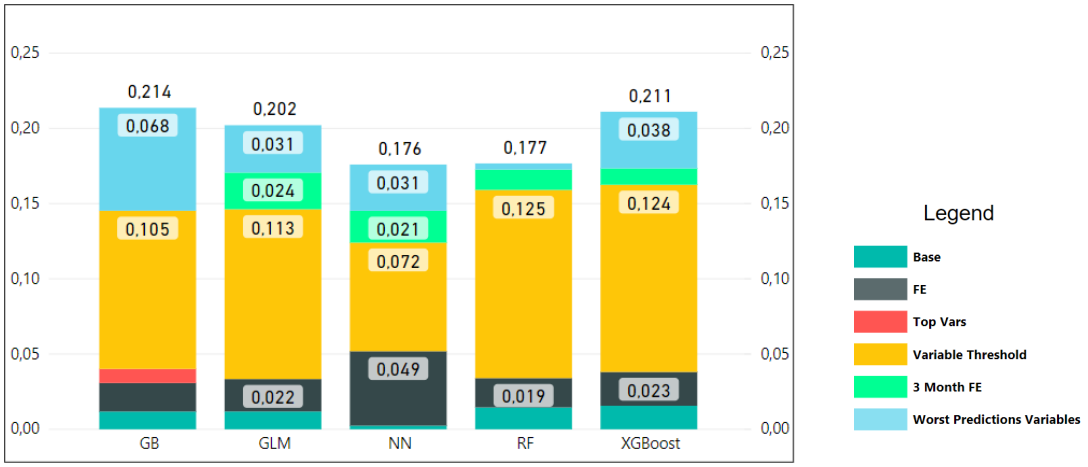


Figure 4.9: Results with all Relevant Variables

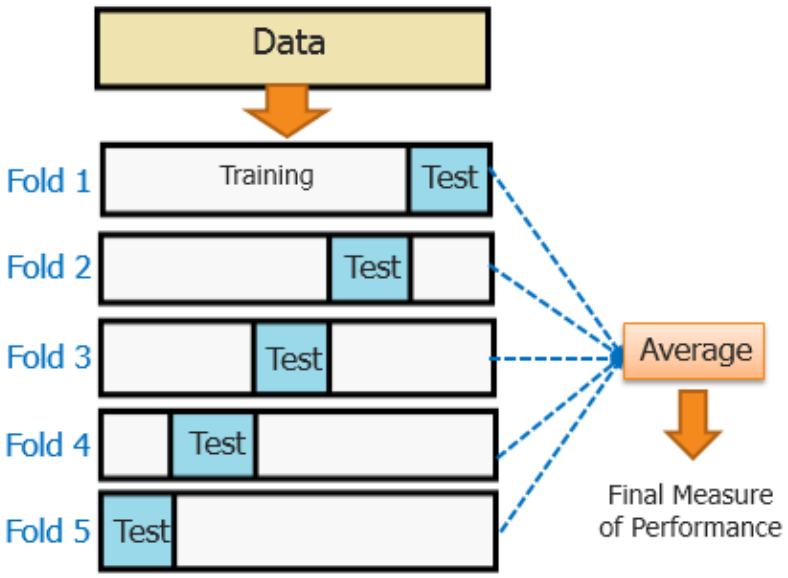


Figure 4.10: K-fold Cross Validation Example (The Tech Check, 2018)

running all the combinations, it selects the best one and applies it to the test set. For example, if for one algorithm 3 parameters are chosen for varying and each of them between 4 different values, 64 combinations (4x4x4) are tested. The parameters altered as well as the respective algorithms can be found in Table 4.1.

The Hyperparameters have the following purpose (LeDell et al., 2019)(Chen et al., 2019):

- Ntrees: Defines the number of decision trees to be used in the algorithm. Varied between 20, 50 and 100;
- eta: Defines the learning rate of the algorithm, scales the contribution of each tree by a factor between 0 and 1. It is used to prevent overfitting by making the Boosting process more conservative. Varied between 0.05, 0.1 and 0.2;
- max_depth: Maximum depth of a tree, the number of decision layers. Varied between 2, 5 and 8;
- min_rows: Fewest allowed observations in a leaf. Varied between 2, 5 and 8 for GB and 5, 15 and 25 for RF;
- hidden_layers: Defines the number of hidden layers as well as the number of neurons in each one. Varied between two layers of 20, 50 and 100 neurons;
- epochs: Number of times the dataset should be iterated. Varied between 10, 50 and 100;
- stopping_rounds: Early stopping based on convergence. Stops the algorithm if the stopping metric does not improve for k rounds. Varied between 5, 20 and 50;
- alpha: Regularization between the Lasso and Ridge penalties. 0 equals to a Ridge regression and a value of 1 produces a Lasso regression. Anything in a middle outputs a mixture of both. Varied between 0 and 1, by increments of 0.1.

During the grid search several metrics were saved in order to have a better idea of the results, the performance metrics in both the test set and the training set, it is expected that in the training set the performance is better, but if there is a big difference to the test set, it means that the parameters are overfitting and can probably be improved. The time taken to apply the model to the test data has also been taken into consideration. The results obtained after the Hyperparameter optimization process are presented in Figure 4.11 (Detailed Results in Appendix C Figure C.1).

The algorithm which presented the biggest improvement was the RF one, which was expected since it was performing poorly in comparison with some theoretically weaker algorithms, like the GLM, and it was likely the its parameters were not optimized. The XGBoost and the GLM also

Table 4.1: Hyperparameters tested per Algorithm

		Algorithm				
		XGBoost	Gradient Boost	Random Forest	Neural Networks	Generalized Liner Model
Parameter	Ntrees	■	■	■		
	eta	■				
	max_depth	■	■	■		
	min_rows		■	■		
	hidden layers				■	
	epochs				■	
	stopping_rounds				■	
	alpha					■

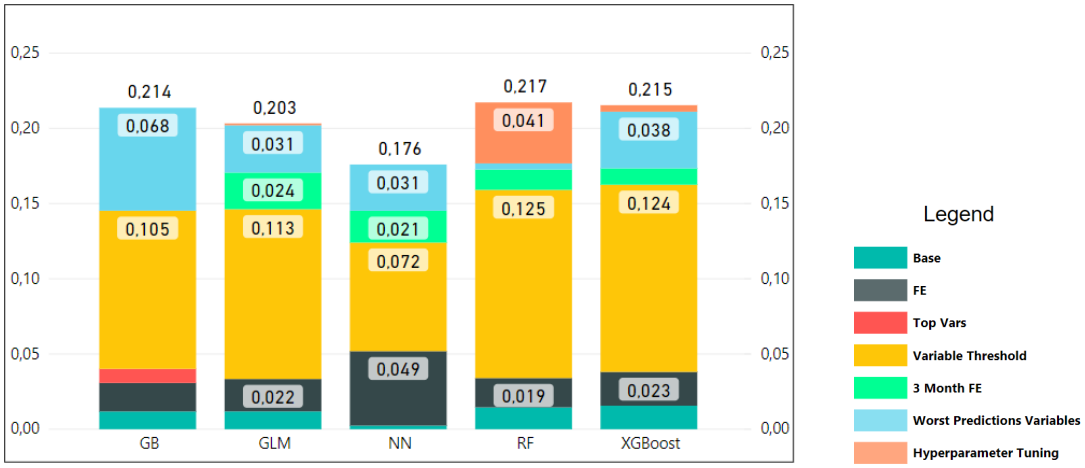
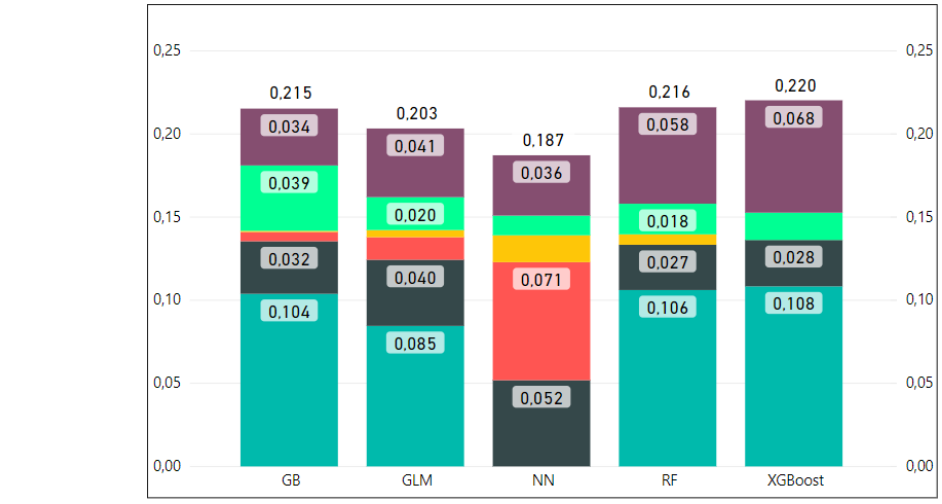


Figure 4.11: Results after Hyperparameter Tuning

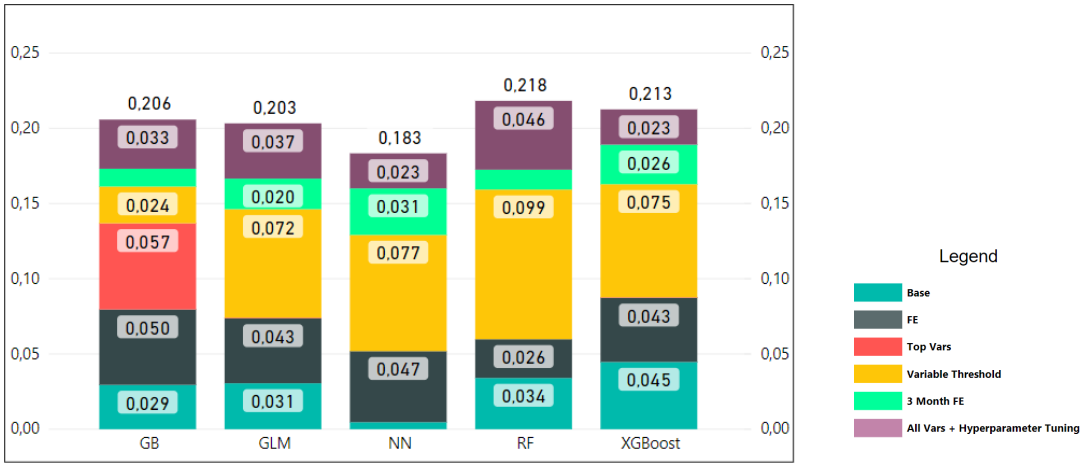
improved, marginally, their performance. Both the NN and the GB performances were better with the default parameters.

Since this is the last step of the optimization process and the data is extremely unbalanced, it is now time to train the model again using class balancing techniques. Given that, a grid search was also developed, varying the same parameters, but including a variable under or oversampling factor. In contrast to the balancing techniques used previously, the undersampled models will not balance the data to a 50/50 distribution, but will instead keep a certain percentage of the negative samples. A variable determining this percentage will vary within the grid between the values of 0.1, 0.3 and 0.5. For the oversampling, there will also be a parameter, varying between the values of 1.5, 3 and 5, that will ensure that the number of positive samples is multiplied by that factor. The NN algorithm will probably benefit the most from this procedure, since it usually does not deal very well with unbalanced domains. The results are represented in Figure 4.12 (Detailed Results in Appendix C Figure C.2 and Figure C.3).

By using dataset balancing techniques, the algorithms were able to outperform the unbalanced dataset and yield better results, that will be described in more detail in Chapter 5. The improvement of the results is natural since in the data is extremely unbalanced and it is very hard for the algorithms to deal with the situation. Moreover, in Section 4.4.3, the class balancing techniques had already proven that they could outperform the traditional model.



(a) Undersampling



(b) Oversampling

Figure 4.12: Final Results using Balancing Techniques

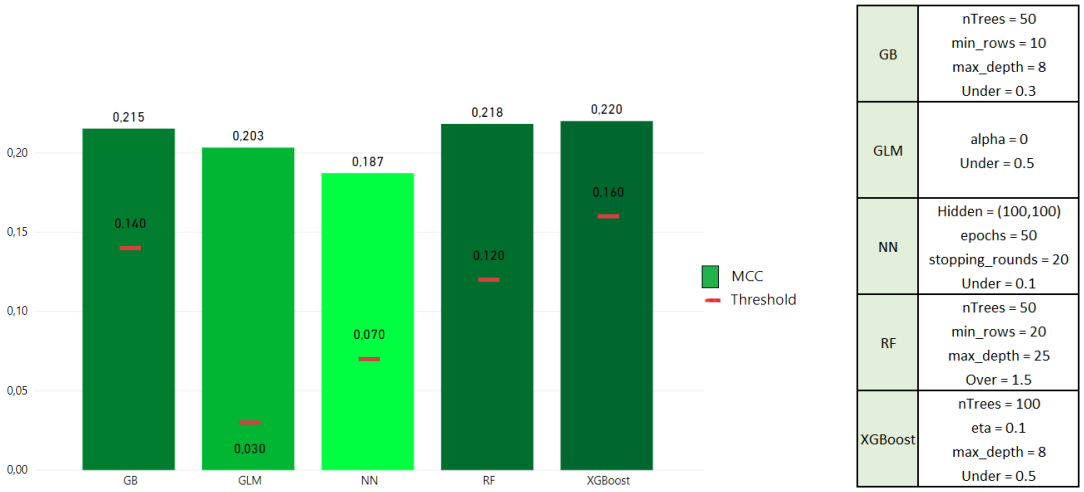
Chapter 5

Results

The CRISP DM process ends with the evaluation of the model in parallel with the business understanding phase so that it can then be moved to the implementation phase. This part of the methodology will be described in the current chapter.

The results obtained show that the current situation at the partner company can be improved and that predicting this type of customer behaviour is possible. The best result for each of the five algorithms, in terms of MCC, is represented in Figure 5.1. The threshold optimized for each of the models is also represented. All of them benefited from data balancing techniques, GB, XGBoost, NN and GLM from undersampling and RF from oversampling. A detailed view of the best sets of hyperparameters can also be seen on Figure 5.1. In terms of MCC, the best performing model was the XGBoost algorithm with an undersampling factor of 0.5.

It is normal that the threshold of every model is lower than 0.5, since the dataset is extremely unbalanced, the probability distribution will be dragged to a lower value and, consequently, the threshold for the positive values will also drop. The tree based models present higher thresholds than the other two because they are capable of dealing with unbalanced data much better.



GB	nTrees = 50 min_rows = 10 max_depth = 8 Under = 0.3
GLM	alpha = 0 Under = 0.5
NN	Hidden = (100,100) epochs = 50 stopping_rounds = 20 Under = 0.1
RF	nTrees = 50 min_rows = 20 max_depth = 25 Over = 1.5
XGBoost	nTrees = 100 eta = 0.1 max_depth = 8 Under = 0.5

Figure 5.1: Best MCC for each model and the respective Parameters

Although the MCC is very relevant to have a quick reference of the model's performance, it is always necessary to look at other indicators. Precision and recall can be very informative, specially when thinking of potential implementations of the model (Equations 5.1 and 5.2). In the context of this project, recall is more important than precision, since the FP are not very clear, in fact, the FP are potential TP, people who did not clearly understand their bill, but that did not contact the partner company about the situation. In contrast, FN should be minimized, since if the objective is to increase customer satisfaction, it is crucial that the partner company is able to identify the majority of the clients who do not understand what they are paying. In Figure 5.2 the values for this two indicators are represented, for each of the models.

$$Recall = \frac{TP}{TP+FN} \tag{5.1}$$

$$Precision = \frac{TP}{TP+FP} \tag{5.2}$$

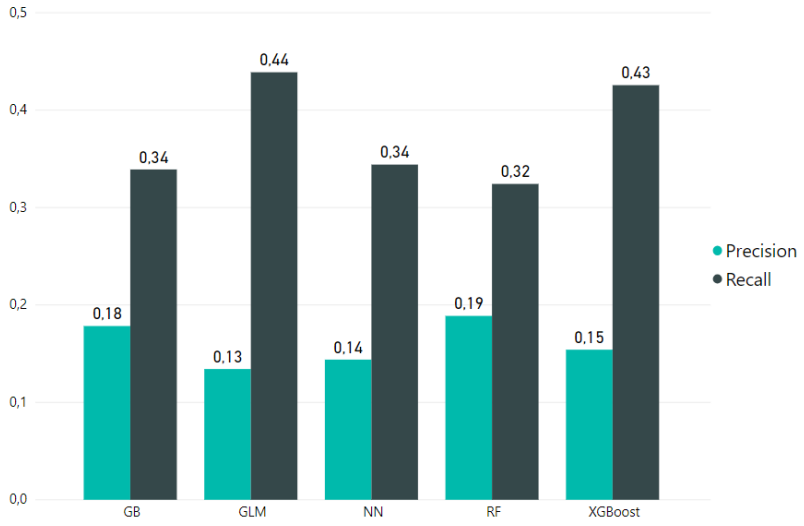


Figure 5.2: Precision and Recall of each model

By analyzing the graph, the algorithms that produce the best results in terms of recall are the XGBoost and the GLM, both having values around 0.43, although the XGBoost has a slightly higher precision. The RF algorithm has the highest value for the precision metric but it is also the lowest in terms of recall. One flaw with this analysis is that it does not take into account the volume of clients that the partner company has. In fact, it is very unrealistic to project a solution that can handle the entire 1,4M clients each month. With that, using the Lift metric, discussed in Section 4.3, can potentially be very useful, by discovering which of the models can fit the largest amount of positive values in just 10% of the data, it becomes easier to distinguish between them (Figure 5.3).

In the first decile of the predictions, the tree based algorithms can fit 14,5% of all the positive

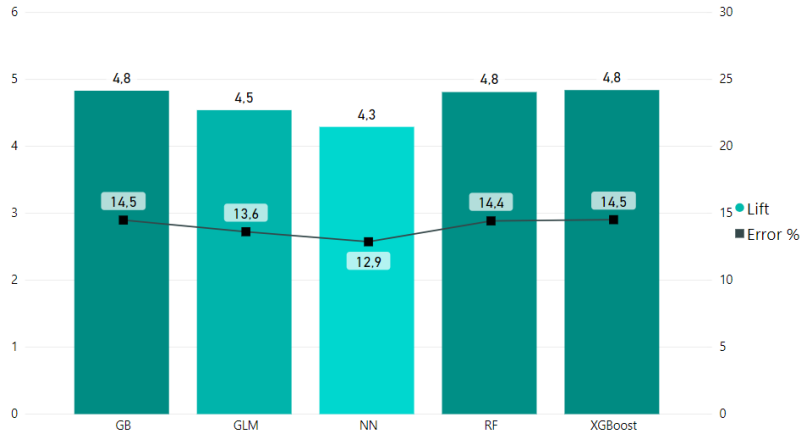


Figure 5.3: Lift and Error Rate for the Top Decile of Predictions

observations, meaning that if the partner company decides to act in this segment of its client base, their success rate will be much higher than in the overall population. Although 10% of the population is the norm for calculating Lift, in this case, since the population is so big, 10% correspond to around 140k clients, which is still a huge number. It is helpful to analyze other percentages of Lift to see what results they would provide. In Figure 5.4, the Lift, for each of the first ten percentiles, for both XGBoost and GB is represented, as well as the total percentage of SR contained in that percentile.

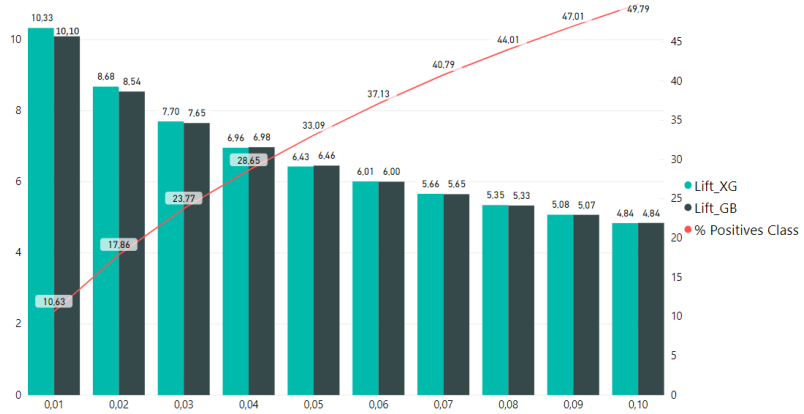


Figure 5.4: Lift and Percentage of the total errors for the first 10 percentiles

It is possible to observe that the two best algorithms in terms of lift are very similar in their behaviour, with XGBoost slightly outperforming GB for the lowest percentiles. As far as the percentage of the total errors, when selecting just 10% of the clients, it is possible to obtain almost half of the total number of billing SR. By analyzing only the first percentile, both algorithms are able to increase the percentage of error more than ten times, increasing from the 3% of the baseline to more than 30%, encompassing around 10% of the total amount of errors in this percentile.

For the beginning of the implementation, it is more reasonable to test the process in 14k people

per month than on 140k and, with that, if the partner company is able to reach 10% of the clients with billing doubts and answer their questions, it will be a good pilot for this implementation that can, in a future step, be implemented in a more automatic way, to ambition reducing the amount of SR by 50%.

If the implementation of the model is to target 1, 10 or 100% of the population, either way the best overall model to be used is the XGBoost with an undersampling factor of 0.5, since it has the highest MCC of all the models, a very high recall in the entire population and the best Lift for smaller segments of the client base.

5.1 Variable Importance

After comparing the models between them and concluding the benefits of using the XGBoost with undersampling, one of the proposed objectives for this project is also learning what causes the clients to not understand their bills clearly. By analyzing what variables of the model influence the most the final outcome it becomes clearer what the problems are (Figure 5.5).

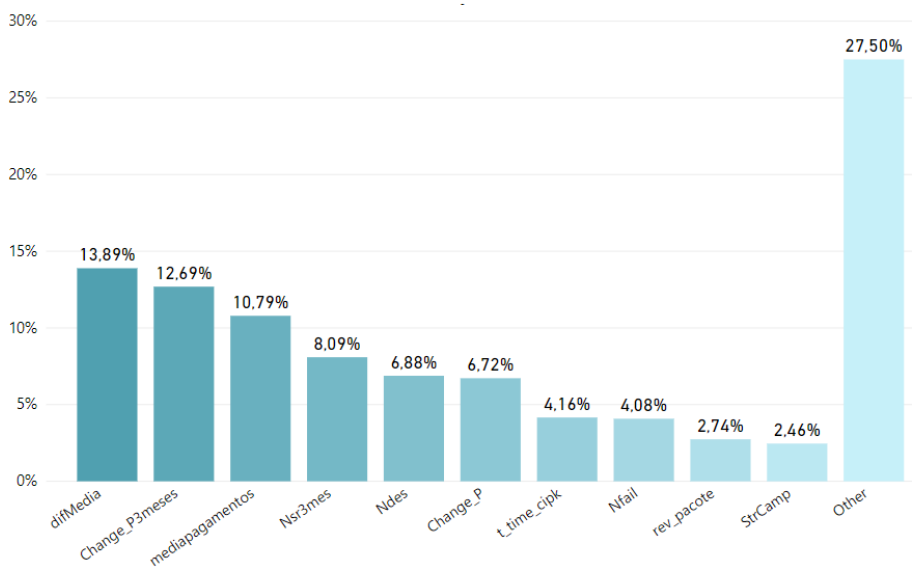


Figure 5.5: Variable Gains for the XGBoost model

In the top 3 most informative variables, two of them are related to the payments in previous months, "mediapagamentos" is the average of the payments in the previous three periods and "difMedia" is the difference in value from the current month's bill and the "mediapagamentos", meaning that changes in these values are very likely to cause dissatisfaction in the client. Changing the bundle, as mentioned in Section 3.4, is also a common factor for generating SR as is represented in variables "Change_P" and "Change_P3meses" that are flags that become positive is the client changed his bundle in the previous month or in the previous three months, respectively. Also mentioned in Section 3.4 was that some clients are more vocal than other and that if one contacted the partner company previously he would be more likely to contact it again. Variable "Nsr3mes"

represents just that, as it is the number of times that a client presented a billing SR in the previous three months. Lastly, the features that were created with the intent of solving the worst predictions (Section 4.4.4), proved to be very relevant for the model in general, since three of them are represented in the top 10 most informative variables. The variables "Ndes" and "Nfail" represent the number of times a client contacted the partner company with the intent of churning or with a service failure and "StrCamp" is a flag that is positive if the client activated a discount campaign in the billing period.

It is interesting to analyze the same scenario of variable importance but, only for certain segments of clients. New clients for example have very different behaviours than long term ones, as Figure 5.6 shows, the variables more important for them are extremely different.

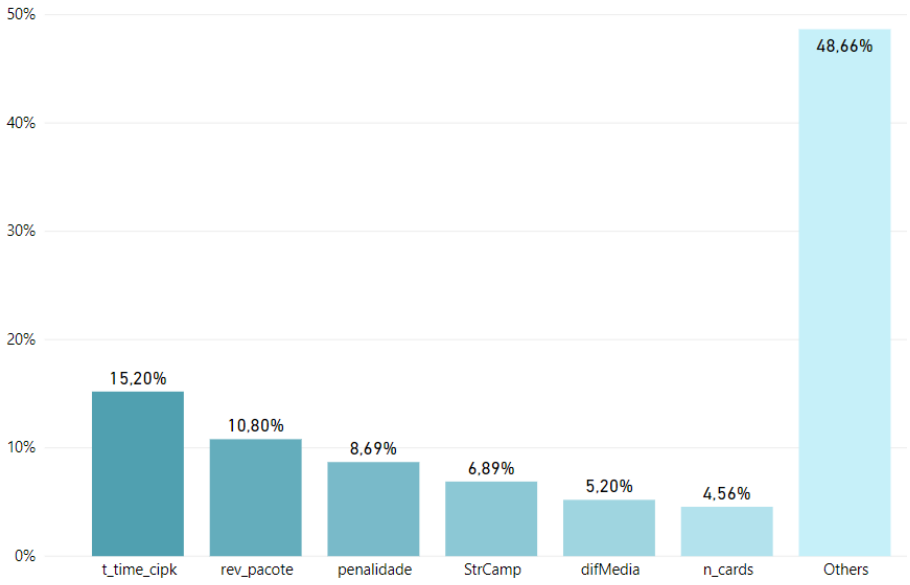


Figure 5.6: Variable Gains for new clients

For this segment of clients the features that are more responsible for billing SR are the time spent using the VOD (Video on Demand) app, the price of the bundle and the penalty for canceling the service before the end of the loyalty period ("t_time_cpk", "rev_pacote" and "penalidade" respectively). All of these variables are more related to the early understanding of the characteristics of the service and not about problems or complaints a client might have.

In Figure 5.7, this analysis is repeated, but for clients who have billing SR frequently, more than two in the previous three months. In this situation, the price paid for the bill is the most frequent reason of complaint.

With this analysis it can be concluded that different segments of clients have different reasons for presenting billing SR and, as such, it can beneficial to segment the dataset and study it separately.

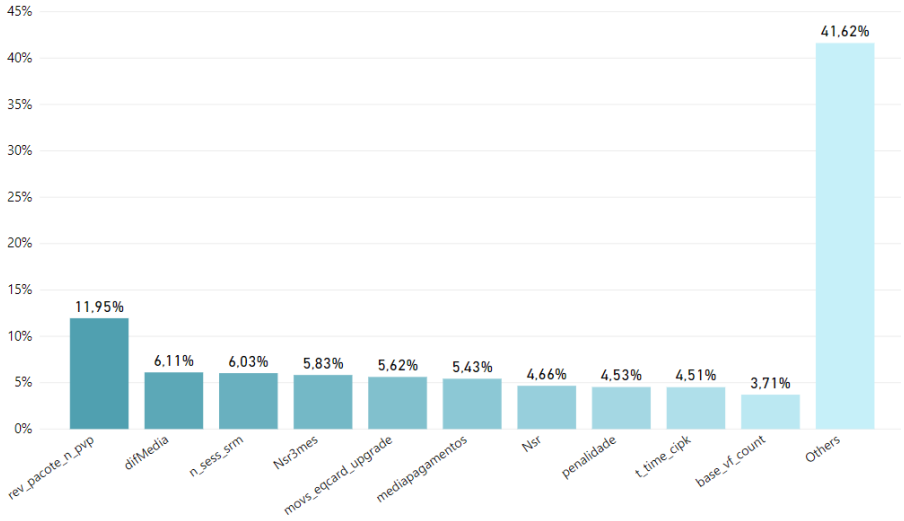


Figure 5.7: Variable Gains for new clients

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The development of this dissertation enabled to answer the initial research questions and draw several conclusions. Most importantly, it was shown that predicting customer behaviour is possible and that, with the right features, it can be done with high confidence. The partner company, where this project was developed, has clear problems in the way they present their bills and, with the model developed, they will have the opportunity of improving the situation and thus enhancing the customers' experience.

The models developed help the partner company identify, with up to ten times more precision, which customers or bills will cause more problems, but it also outputs several False Positive cases. These false positives can not be seen as a complete misclassification, since some of them are, in fact, people who had troubles understanding their bills, but, because they did not contact the partner company due to some other external circumstances, that it is not possible to gather information about, they do not present a SR. The importance of Feature Engineering for a predictive modeling problem like this one should also be highlighted. Of the top ten most informative variables for the model, discussed in Section 5.1, eight of them were a product of FE and, throughout the whole project, the several iterations of this process all proved to be beneficial for the results. One other conclusion that can be drawn is that XGBoost does in fact produce better overall results than the regular GB algorithm, at least when it comes to the entire sample of data, because when observing just a small percentage of the best predictions, the two algorithms proved to be very similar.

In the beginning of the project, three improvements points were identified for the current billing SR handling process. The first one was to reduce the number of billing SR presented by the clients. This improvement point is expected to be fulfilled, since the developed algorithm can identify who these clients are and is now up to the partner company to implement a solution using this information. The second one was to give information to the client in a clearer way. By identifying the variables most responsible for affecting the customers' behaviour of the clients, it is now clear which aspects of the bill cause more confusion. This is very valuable for the company since now it is easier to perceive the client and implement some changes that make it easier for

them to understand what they are paying for. Lastly, the third and final improvement point was to distribute customer service resources, like people and money, more evenly. If the partner company is able to successfully implement the model developed and reduce the amount of billing SR to some degree, they will have room to move some resources and distribute them in a way that is more advantageous.

During the development of the project, several hurdles had to be overcome that should be highlighted. First of all, working with a very large volume of data was very challenging, specially with the limited resources available. The dataset had to be converted into a different format and split among several files in order to be workable. In order to run the algorithms, the computer had to be limited in terms of both processor and memory, slowing down the process, since there were several issues with the computer overheating. Secondly, the Feature Engineering procedure was very intensive, the creation of variables relevant for the model required several hours of brainstorming with the more senior members of the data science team in the partner company, in order to conceptualize what features would be useful to improve the results. After the conceptualization, discovering were the data could be gathered amidst the terabytes of information and hundreds of data tables of the partner company was a big challenge. It was then necessary to combine information from several sources in order to create the features. Lastly, working with an extremely unbalanced dataset like the one in this project proved to be a constant challenge, with several techniques having to be employed to handle the situation, like dynamic threshold and class balancing.

Concluding, the project objectives were successfully accomplished, the several challenges presented were overcome, and when the model is fully implemented it will enable the company to improve the relationship with their customers and, consequently, their satisfaction. Customer satisfaction is critical for all companies and is determinant to decrease customer churn. The model developed can also potentially enable the company to use its resources more effectively.

6.2 Future Work

After the conclusion of the project, it was shown that the model has potential in predicting the behaviour of customers regarding their bills. Although this is the case, there is room for future work in order to go further into the topic.

First of all, using demographic variables, like the customer's age, gender, social status and possibly even their income, could potentially be relevant for the model and prove to be very useful for the final results. Moreover, it could be beneficial to test the model in different time periods. During the duration of this dissertation, it was only possible to study this topic using the months of July and August, due to limitations in the data available, but by observing Figure 6.1, these months correspond to the period of the year with the least amount of SR, probably due to being typical holiday months, and developing the model using other pairs could potentially prove to be beneficial, since the algorithms would have more positive samples to learn from.

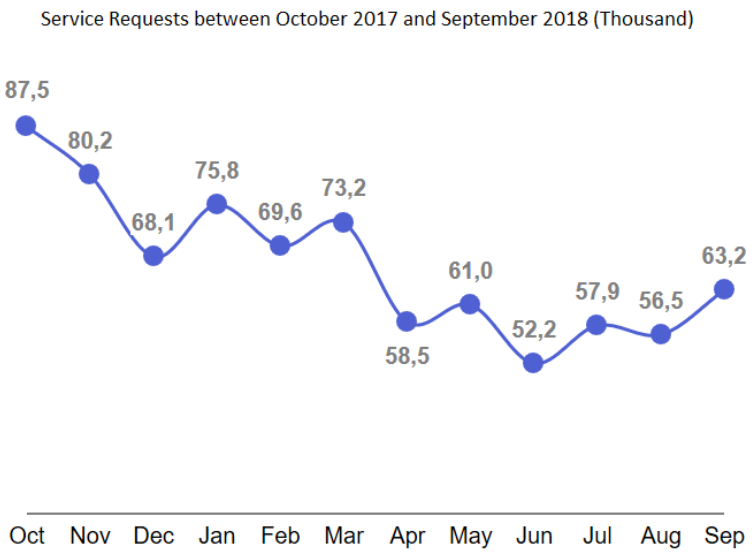


Figure 6.1: Amount of SR each Month (Partner company's internal documents, 2019a)

It would also be interesting to test mechanisms like the sliding window one, where several pairs of train and test set are created, the algorithms are tested in each set and the results averaged. This mechanism has proven to be beneficial in terms of reducing the standard deviation associated with each prediction but it is known to be very resource intensive, although there are already attempts to optimize this procedure, like the one proposed by Wojek et al. (2008). There are also some more complex class balancing techniques, like SMOTE, as mentioned in Section 2.3.2, that could prove to be an improvement over the standard ones, as well as a larger oversampling factor that would take the data closer to balance. Another method to improve the results would be to do a more thorough hyperparameter tuning, since the grid search that was employed, due to time and technological constraints, was not very extensive.

In the results section of this dissertation, it was concluded that different segments of clients have very distinct responses to the model, and that by modeling the client base by segments instead of the entire population at once, could prove to be an improvement to the model developed.

Some of the billing SR are a symptom of billing errors by the partner company and since those are not related to the customer, they will always be harder to predict using the current dataset. With that being said, if another project was to be initiated with the intent of minimizing these errors, it would benefit a lot the current project, since the algorithms would have clearer positive samples.

Lastly, measuring the actual impact of the project would be interesting. The project needed to be fully implemented within the system and then some conclusions could be drawn, if the NPS increased, if the number of billing SR decreased and if it had any implications in terms of costs for the company.

Bibliography

Almossawi, M. M. (2012). Customer satisfaction in the mobile telecom industry in bahrain: Antecedents and consequences. *International Journal of Marketing Studies*, 4(6):139.

Arcuri, A. and Fraser, G. (2013). Parameter tuning or default values? an empirical investigation in search-based software engineering. *Empirical Software Engineering*, 18(3):594–623.

Azevedo, A. I. R. L. and Santos, M. F. (2008). Kdd, semma and crisp-dm: a parallel overview. *IADS-DM*.

Azzarello, D. and Kovac, M. (2011). Can communications service providers earn their customers' love? *Bain and Company, August*.

Bhargava, N., Sharma, G., Bhargava, R., and Mathuria, M. (2013). Decision tree analysis on j48 algorithm for data mining. *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering*, 3(6).

Bolton, R. N. and Bronkhorst, T. M. (1995). The relationship between customer complaints to the firm and subsequent exit behavior. *ACR North American Advances*.

Branco, P., Torgo, L., and Ribeiro, R. P. (2016). A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):31.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., and Li, Y. (2019). *xgboost: Extreme Gradient Boosting*. R package version 0.82.1.

David, A., Dickey, N., Carolina State, U., and Raleigh, N. C. (2012). Introduction to predictive modelling with examples. *SAS Global Forum*.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., et al. (1996). Knowledge discovery and data mining: Towards a unifying framework. In *KDD*, volume 96, pages 82–88.

from Jed Wing, M. K. C., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., Mayer, Z., Kenkel, B., the R Core Team, Benesty, M., Lescarbeau, R., Ziem, A., Scrucca, L., Tang, Y., Candan, C., and Hunt., T. (2018). *caret: Classification and Regression Training*. R package version 6.0-81.

Grothendieck, G. (2017). *sqldf: Manipulate R Data Frames Using SQL*. R package version 0.4-11.

Hadden, J., Tiwari, A., Roy, R., and Ruta, D. (2006). Churn prediction using complaints data. In *Proceedings of world academy of science, engineering and technology*. Citeseer.

Haring, M., Offermann, S., Danker, T., Horst, I., Peterhansel, C., and Stam, M. (2007). Chromatin immunoprecipitation: optimization, quantitative analysis and data normalization. *Plant methods*, 3(1):11.

Jung, Y. and Hu, J. (2015). Ak-fold averaging cross-validation procedure. *Journal of nonparametric statistics*, 27(2):167–179.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

LeDell, E., Gill, N., Aiello, S., Fu, A., Candel, A., Click, C., Kraljevic, T., Nykodym, T., Aboyoun, P., Kurka, M., and Malohlava, M. (2019). *h2o: R Interface for 'H2O'*. R package version 3.22.1.1.

Lemon, K. N. and Verhoef, P. C. (2016). Understanding customer experience throughout the customer journey. *Journal of marketing*, 80(6):69–96.

Ling, C. X., Huang, J., Zhang, H., et al. (2003). Auc: a statistically consistent and more discriminating measure than accuracy. In *Ijcai*, volume 3, pages 519–524.

Liu, Y., Cheng, J., Yan, C., Wu, X., and Chen, F. (2015). Research on the matthews correlation coefficients metrics of personalized recommendation algorithm evaluation. *International Journal of Hybrid Information Technology*, 8(1):163–172.

Margineantu, D. D. and Dietterich, T. G. (1997). Pruning adaptive boosting. In *ICML*, volume 97, pages 211–218.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*. Chapman & Hall / CRC, London.

Natekin, A. and Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21.

Open Data Science (2018). Logistic regression. [Online; accessed June 8, 2019].

Oreski, D., Oreski, S., and Klicek, B. (2017). Effects of dataset characteristics on the performance of feature selection techniques. *Applied Soft Computing*, 52:109–119.

Otaris (2018). Data analysis, modeling and reporting. [Online; accessed April 2, 2019].

Partner company's internal documents (2019a). Billing service requests.

Partner company's internal documents (2019b). Customer perception of essential elements.

Roos, I. and Edvardsson, B. (2008). Customer-support service in the relationship perspective. *Managing Service Quality: An International Journal*, 18(1):87–107.

Schwartz, L. R. and Overton, D. T. (1987). Emergency department complaints: a one-year analysis. *Annals of emergency medicine*, 16(8):857–861.

Singh, Y. and Chauhan, A. S. (2009). Neural networks in data mining. *Journal of Theoretical & Applied Information Technology*, 5(1).

Sloo, M. A. (1999). Method and apparatus for handling complaints. US Patent 5,895,450.

Sola, J. and Sevilla, J. (1997). Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on nuclear science*, 44(3):1464–1468.

StackExchange (2011). Lift measure in data mining. [Online; accessed April 27, 2019].

The Tech Check (2018). Different types of validations in machine learning (cross validation). [Online; accessed May 8, 2019].

Touw, W. G., Bayjanov, J. R., Overmars, L., Backus, L., Boekhorst, J., Wels, M., and van Hijum, S. A. (2012). Data mining in the life sciences with random forest: a walk in the park or lost in the jungle? *Briefings in bioinformatics*, 14(3):315–326.

Vera Miguéis (2018a). Pre processing. [Online; accessed February 20, 2019].

Vera Miguéis (2018b). Prediction models evaluation. [Online; accessed February 27, 2019].

Wojek, C., Dorkó, G., Schulz, A., and Schiele, B. (2008). Sliding-windows for rapid object class localization: A parallel technique. In *Joint Pattern Recognition Symposium*, pages 71–81. Springer.

Wu, C., Chau, K., and Li, Y. (2009). Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques. *Water Resources Research*, 45(8).

Appendix A

Results of the several iterations

Table A.1: Base Results for the Model

			Confusion Matrix				Performance Indicators					
	Model	Balance technique	TP	TN	FP	FN	Accuracy	Precision	Recall	Fscore - Test	Lift - Test (1%)	MCC - Test
Base	GLM	None	22	1364131	49	40385	0,971	0,310	0,001	0,001	4,650	0,012
	RF	None	44	1364087	93	40363	0,971	0,321	0,001	0,002	6,090	0,017
	GradientB	None	19	1364136	44	40388	0,971	0,302	0,000	0,001	5,750	0,011
	NN	None	1	1364180	0	40406	0,971	1,000	0,000	0,000	2,240	0,005
	XGBoost	None	25	1364095	85	40382	0,971	0,227	0,001	0,001	5,690	0,011
	GLM	Under	25450	841047	523133	14957	0,617	0,046	0,630	0,086	4,500	0,084
	RF	Under	28205	836326	527854	12202	0,616	0,051	0,698	0,095	4,810	0,106
	GradientB	Under	28047	843413	520767	12360	0,620	0,051	0,694	0,095	5,250	0,107
	NN	Under	40407	0	1364180	0	0,029	0,029	1,000	0,056	2,020	
	XGBoost	Under	27284	865030	499150	13123	0,635	0,052	0,675	0,096	5,180	0,107
	GLM	Over	225	1363289	891	40182	0,971	0,202	0,006	0,011	4,630	0,029
	RF	Over	247	1363402	778	40160	0,971	0,241	0,006	0,012	5,950	0,034
	GradientB	Over	332	1363121	1059	40075	0,971	0,239	0,008	0,016	5,740	0,040
	NN	Over	1	1364178	2	40406	0,971	0,333	0,000	0,000	3,140	0,003
	XGBoost	Over	410	1362592	1588	39997	0,970	0,205	0,010	0,019	5,570	0,040

Table A.2: Results after the initial Feature Engineering

			Confusion Matrix				Performance Indicators					
	Model	Balance technique	TP	TN	FP	FN	Accuracy	Precision	Recall	Fscore - Test	Lift - Test (1%)	MCC - Test
FE	GLM	None	169	1363823	357	40238	0,971	0,321	0,004	0,008	6,610	0,034
	RF	None	101	1364024	156	40306	0,971	0,393	0,002	0,005	7,150	0,029
	GradientB	None	128	1363952	228	49279	0,965	0,360	0,003	0,005	7,690	0,028
	NN	None	60	1364063	117	40347	0,971	0,339	0,001	0,003	5,450	0,021
	XGBoost	None	149	1363892	288	40258	0,971	0,341	0,004	0,007	7,450	0,033
	GLM	Under	23222	1024395	339785	17185	0,746	0,064	0,575	0,115	6,130	0,124
	RF	Under	26070	979934	384246	14337	0,716	0,064	0,645	0,116	6,440	0,134
	GradientB	Under	25850	996999	367181	14557	0,728	0,066	0,640	0,119	7,190	0,138
	NN	Under	40407	0	1364180	0	0,029	0,029	1,000	0,056	4,750	
	XGBoost	Under	25979	988706	375474	14428	0,722	0,065	0,643	0,118	6,770	0,136
	GLM	Over	1376	1358869	5311	39031	0,968	0,206	0,034	0,058	6,520	0,073
	RF	Over	601	1362536	1644	39806	0,970	0,268	0,015	0,028	6,970	0,057
	GradientB	Over	1498	1359878	4302	38909	0,969	0,258	0,037	0,065	7,530	0,088
	NN	Over	0	1364179	1	40407	0,971	0,000	0,000		5,910	0,000
	XGBoost	Over	1418	1359946	4234	38989	0,969	0,251	0,035	0,062	7,520	0,084

Table A.3: Results with the Top variables

			Confusion Matrix				Performance Indicators					
	Model	Balance technique	TP	TN	FP	FN	Accuracy	Precision	Recall	Fscore - Test	Lift - Test (1%)	MCC - Test
Top Variables	GLM	None	166	1363831	349	40241	0,971	0,322	0,004	0,008	6,600	0,034
	RF	None	99	1364021	159	40308	0,971	0,384	0,002	0,005	6,930	0,029
	GradientB	None	128	1363961	219	40279	0,971	0,369	0,003	0,006	7,630	0,032
	NN	None	130	1363738	442	40277	0,971	0,227	0,003	0,006	4,680	0,024
	XGBoost	None	157	1363880	300	40250	0,971	0,344	0,004	0,008	7,420	0,034
	GLM	Under	23316	1019731	344449	17091	0,743	0,063	0,577	0,114	6,130	0,123
	RF	Under	26225	975358	388822	14182	0,713	0,063	0,649	0,115	6,160	0,133
	GradientB	Under	25893	990638	373542	14514	0,724	0,065	0,641	0,118	7,130	0,136
	NN	Under	40407	0	1364180	0	0,029	0,029	1,000	0,056	2,980	
	XGBoost	Under	26138	981619	382561	14269	0,717	0,064	0,647	0,116	6,820	0,135
	GLM	Over	1390	1358804	5376	39017	0,968	0,205	0,034	0,059	6,480	0,074
	RF	Over	537	1362718	1462	39870	0,971	0,269	0,013	0,025	6,900	0,054
	GradientB	Over	1452	1359966	4214	38955	0,969	0,256	0,036	0,063	7,520	0,087
	NN	Over	1	1364180	0	40406	0,971	1,000	0,000	0,000	5,250	0,005
	XGBoost	Over	1619	1359037	5143	38788	0,969	0,239	0,040	0,069	7,310	0,088

Table A.4: Results with the Variable Threshold

			Confusion Matrix				Performance Indicators					
	Model	Balance technique	TP	TN	FP	FN	Accuracy	Precision	Recall	Fscore - Test	Lift - Test (1%)	MCC - Test
Variable Threshold	GLM	None	14048	1244117	120063	26321	0,896	0,105	0,348	0,161	6,600	0,148
	RF	None	3045	1351965	12215	37362	0,965	0,200	0,075	0,109	7,110	0,107
	GradientB	None	12220	1282074	82106	28187	0,921	0,130	0,302	0,181	7,630	0,162
	NN	None	8485	1286038	78142	31922	0,922	0,098	0,210	0,134	5,200	0,106
	XGBoost	None	11445	1290542	73638	28962	0,927	0,135	0,283	0,182	7,420	0,161
	GLM	Under	19195	1143509	220671	21212	0,828	0,080	0,475	0,137	6,160	0,139
	RF	Under	23096	1071659	292521	17311	0,779	0,073	0,572	0,130	6,170	0,143
	GradientB	Under	24122	1049541	314639	16285	0,764	0,071	0,597	0,127	7,240	0,143
	NN	Under	16889	1125533	238647	23518	0,813	0,066	0,418	0,114	5,830	0,105
	XGBoost	Under	35699	545327	818853	4708	0,414	0,042	0,883	0,080	6,960	0,097
	GLM	Over	14722	1234950	129230	25685	0,890	0,102	0,364	0,160	6,480	0,149
	RF	Over	3318	1349671	14509	37089	0,963	0,186	0,082	0,114	6,720	0,107
	GradientB	Over	12831	1275759	88421	27576	0,917	0,127	0,318	0,181	7,520	0,163
	NN	Over	12198	1254432	109748	28209	0,902	0,100	0,302	0,150	5,290	0,131
	XGBoost	Over	14874	1248016	116164	25533	0,899	0,114	0,368	0,174	7,310	0,163

Table A.5: Results with the more Feature Engineering

			Confusion Matrix				Performance Indicators					
	Model	Balance technique	TP	TN	FP	FN	Accuracy	Precision	Recall	Fscore - Test	Lift - Test (1%)	MCC - Test
3 Month FE	GLM	None	21062	1153751	210429	19345	0,836	0,091	0,521	0,155	7,490	0,165
	RF	None	3768	1352155	12025	36639	0,965	0,239	0,093	0,134	8,520	0,134
	GradientB	None	15549	1259715	104465	24858	0,908	0,130	0,385	0,194	8,750	0,184
	NN	None	17937	1178735	185445	22470	0,852	0,088	0,444	0,147	6,350	0,146
	XGBoost	None	12376	1293521	70659	28031	0,930	0,149	0,306	0,201	8,640	0,180
	GLM	Under	24103	1092244	271936	16304	0,795	0,081	0,597	0,143	7,260	0,163
	RF	Under	23106	1144567	219613	17301	0,831	0,095	0,572	0,163	7,240	0,182
	GradientB	Under	28302	1027305	336875	12105	0,752	0,078	0,700	0,140	7,970	0,173
	NN	Under	31469	283354	1080826	8938	0,224	0,028	0,779	0,055	2,130	-0,006
	XGBoost	Under	33795	779131	585049	6612	0,579	0,055	0,836	0,103	7,610	0,137
	GLM	Over	22225	1133510	230670	18182	0,823	0,088	0,550	0,152	7,490	0,166
	RF	Over	6433	1337111	27069	33974	0,957	0,192	0,159	0,174	8,160	0,153
	GradientB	Over	20384	1195140	169040	20023	0,865	0,108	0,504	0,177	8,720	0,186
	NN	Over	22129	1132233	231947	18278	0,822	0,087	0,548	0,150	7,160	0,164
	XGBoost	Over	16884	1244803	119377	23523	0,898	0,124	0,418	0,191	8,610	0,187

Appendix B

Detailed Results of Individual Variables

Table B.1: Results with Churn Requests Information

		Confusion Matrix				Performance Indicators						
DataSet	Model	TP	TN	FP	FN	Accuracy	Precision	Recall	Threshold	Fscore	Lift (1%)	MCC
Churn Requests	GLM	17875	1230656	133524	22532	0,889	0,118	0,442	0,070	0,186	8,630	0,186
	RF	4234	1351910	12270	36173	0,966	0,257	0,105	0,220	0,149	9,230	0,149
	GradientB	17211	1254458	109722	23196	0,905	0,136	0,426	0,080	0,206	9,590	0,201
	NN	12888	1243789	120391	27519	0,895	0,097	0,319	0,070	0,148	6,350	0,132
	XGBoost	12089	1305564	58616	28318	0,938	0,171	0,299	0,110	0,218	9,540	0,196

Table B.2: Results with Maintenance Information

		Confusion Matrix				Performance Indicators						
DataSet	Model	TP	TN	FP	FN	Accuracy	Precision	Recall	Threshold	Fscore	Lift (1%)	MCC
Maintnace Interventions	GLM	21281	1150365	213815	19126	0,834	0,091	0,527	0,040	0,154	7,510	0,166
	RF	3295	1353567	10613	37112	0,966	0,237	0,082	0,220	0,121	8,230	0,125
	GradientB	17781	1232101	132079	22626	0,890	0,119	0,440	0,070	0,187	8,710	0,186
	NN	17201	1167439	196741	23206	0,843	0,080	0,426	0,020	0,135	5,130	0,131
	XGBoost	12414	1292794	71386	27993	0,929	0,148	0,307	0,100	0,200	8,690	0,180

Table B.3: Results with Service Failure Information

		Confusion Matrix				Performance Indicators						
DataSet	Model	TP	TN	FP	FN	Accuracy	Precision	Recall	Threshold	Fscore	Lift (1%)	MCC
Service Failures	GLM	15257	1260801	103379	25150	0,908	0,129	0,378	0,070	0,192	8,570	0,181
	RF	4136	1351187	12993	36271	0,965	0,241	0,102	0,220	0,144	8,790	0,141
	GradientB	16828	1257744	106436	23579	0,907	0,137	0,416	0,080	0,206	9,540	0,200
	NN	19287	1119754	244426	21120	0,811	0,073	0,477	0,070	0,127	5,410	0,128
	XGBoost	12100	1303579	60601	28307	0,937	0,166	0,299	0,110	0,214	9,310	0,192

Table B.4: Results with Discount Campaigns Information

		Confusion Matrix				Performance Indicators						
DataSet	Model	TP	TN	FP	FN	Accuracy	Precision	Recall	Threshold	Fscore	Lift (1%)	MCC
Discount Campaigns	GLM	20958	1169296	194884	19449	0,847	0,097	0,519	0,040	0,164	7,810	0,174
	RF	4928	1348626	15554	35479	0,964	0,241	0,122	0,210	0,162	9,230	0,154
	GradientB	15099	1271645	92535	25308	0,916	0,140	0,374	0,090	0,204	9,300	0,192
	NN	17174	1186291	177889	23233	0,857	0,088	0,425	0,010	0,146	5,250	0,142
	XGBoost	11575	1306073	58107	28832	0,938	0,166	0,286	0,110	0,210	9,310	0,188

Table B.5: Results with all the Variables except Maintenance

		Confusion Matrix				Performance Indicators						
DataSet	Model	TP	TN	FP	FN	Accuracy	Precision	Recall	Threshold	Fscore	Lift (1%)	MCC
Worst Predictions Variables	GLM	18196	1242968	121212	22211	0,898	0,131	0,450	0,060	0,202	9,280	0,202
	RF	5717	1348360	15820	34690	0,964	0,265	0,141	0,220	0,185	10,330	0,177
	GradientB	13635	1300798	63382	26772	0,936	0,177	0,337	0,110	0,232	10,460	0,214
	NN	20789	1174667	189513	19618	0,851	0,099	0,514	0,050	0,166	6,580	0,176
	XGBoost	12545	1309180	55000	27862	0,941	0,186	0,310	0,120	0,232	10,360	0,211

Appendix C

Hyperparameter Tuning Results

Table C.1: Results after Hyperparameter Tuning

			Confusion Matrix				Performance Indicators							
DataSet	Model	Balance Technique	TP	TN	FP	FN	Accuracy	Precision	Recall	Threshold	Fscore	Lift (1%)	MCC	Hyperparameters
Parameters	GLM	None	17876	1248031	116149	22531	0,901	0,133	0,442	0,030	0,205	4,540	0,203	alpha=0
	RF	None	12402	1313273	50907	28005	0,944	0,196	0,307	0,130	0,239	4,850	0,217	ntrees=100;min_rows=20;max_depth=25
	GradientB	None	7672	1339741	24439	32735	0,959	0,239	0,190	0,170	0,212	4,820	0,192	ntrees=50;min_rows=10;max_depth=8
	NN	None	21187	1128236	235944	19220	0,818	0,082	0,524	0,050	0,142	3,800	0,152	Hidden={100,100};epochs=100;stopping_rounds=50
	XGBoost	None	12601	1310767	53413	27806	0,942	0,191	0,312	0,120	0,237	4,840	0,215	nrounds=100;eta=0.1;max_depth=8

Table C.2: Results after Hyperparameter Tuning and Undersampling

			Confusion Matrix				Performance Indicators							
DataSet	Model	Balance Technique	TP	TN	FP	FN	Accuracy	Precision	Recall	Threshold	Fscore	Lift (1%)	MCC	Hyperparameters
Parameters Under	GLM	Under	17734	1249806	114374	22673	0,902	0,134	0,439	0,030	0,206	4,540	0,203	alpha=0 ; Under = 0.5
	RF	Under	12069	1315290	48890	28338	0,945	0,198	0,299	0,130	0,238	4,840	0,216	ntrees=100;min_rows=10;max_depth=25; Under=0.5
	GradientB	Under	13702	1301062	63118	26705	0,936	0,178	0,339	0,140	0,234	4,830	0,215	ntrees=50;min_rows=10;max_depth=8; Under=0.3
	NN	Under	13904	1281452	82728	26503	0,922	0,144	0,344	0,070	0,203	4,290	0,187	Hidden={100,100,100};epochs=50;stopping_rounds=20; Under=0.1
	XGBoost	Under	17199	1260752	94428	23208	0,916	0,154	0,426	0,160	0,226	4,840	0,220	nrounds=100;eta=0.1;max_depth=8; Under=0.5

Table C.3: Results after Hyperparameter Tuning and Oversampling

			Confusion Matrix				Performance Indicators							
DataSet	Model	Balance Technique	TP	TN	FP	FN	Accuracy	Precision	Recall	Threshold	Fscore	Lift (1%)	MCC	Hyperparameters
Parameters Over	GLM	Over	17876	1248031	116149	22531	0,901	0,133	0,442	0,030	0,205	4,540	0,203	alpha=0 ; Over =1.5
	RF	Over	13102	1307873	56307	27305	0,940	0,189	0,324	0,120	0,239	4,810	0,218	ntrees=50;min_rows=20;max_depth=25; Over=1.5
	GradientB	Over	10014	1327420	36760	30393	0,952	0,214	0,248	0,140	0,230	4,830	0,206	ntrees=100;min_rows=50;max_depth=8; Over=3
	NN	Over	16986	1240244	123936	23421	0,895	0,121	0,420	0,080	0,187	4,190	0,183	Hidden={100,100,100};epochs=50;stopping_rounds=50; Over=1.5
	XGBoost	Over	12173	1313054	51126	28234	0,943	0,192	0,301	0,170	0,235	4,820	0,213	nrounds=100;eta=0.1;max_depth=8; Over=1.5