FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Churn Prediction in the Telecom Business

Georgina Cunha Esteves

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: João Pedro Mendes Moreira

June 25, 2016

Churn Prediction in the Telecom Business

Georgina Cunha Esteves

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Doctor Name of the President
External Examiner: Doctor Name of the Examiner
Supervisor: João Pedro Mendes Moreira

June 25, 2016

Abstract

Telecommunication companies are acknowledging the existing connection between customer satisfaction and company revenues. Customer churn in telecom refers to a customer that ceases his relationship with a company. Besides losing the customer, it is also likely that the customer will join a competitor company. These companies rely on three main strategies to generate more revenue: acquire new customers, upsell the existing ones or increase customer retention. Some articles suggest that churn prediction in telecom has recently gained substantial interest of stakeholders, who noticed that retaining a customer is substantially cheaper that gaining a new one. This dissertation compares six approaches that identify the clients who are closer to abandon their telecom provider. Those algorithms are: KNN, Naive Bayes, J48, Random Forest, AdaBoost and ANN. For the purpose of this research, real data was provided by WeDo technologies, which is the number one provider of revenue assurance and fraud management software for telecom operators. The use of real data extended the refinement time necessary, but ensured that the developed algorithm and model can be applied to real world situations. The large dataset available opens a new set of possibilities, and makes it possible to obtain interesting and novel results. The models were evaluated according to three criteria: area under curve, sensitivity and specificity, with special weight to the first two values. The Random Forest algorithm proved to be the most adequate in all the test cases.

Keywords: Churn Prediction; Telecom; Machine Learning; Churn Analysis; Customer attritions analysis; Customer attritions.

Resumo

As empresas na área das telecomunicações estão a aperceber-se da forte ligação existente entre a satisfação do cliente e as receitas da empresa. Churn de clientes na área das telecomunicações refere-se a um cliente que cessa o contrato mantido com uma empresa. Além de perder o cliente, é também provável que o cliente irá juntar-se de uma empresa concorrente. As empresas contam com três estratégias principais para gerar mais receita: adquirir novos clientes, vender um novo serviço a clientes da empresa ou aumentar a retenção de clientes. Alguns artigos sugerem que a previsão de churn na áre das telecomunicações ganhou recentemente interesse substancial por parte das empresas, que notaram que reter um cliente é substancialmente mais barato do que adquirir um novo. Esta dissertação compara seis abordagens que identificam os clientes que estão mais perto de abandonar o seu fornecedor de telecomunicações. Esses algoritmos são: KNN, Naive Bayes, C4.5, Random Forest, AdaBoost e ANN. Para a realização desta pesquisa foram-nos fornecidos dados de clientes reais pela Wedo technologies, uma empresa que se especializou no fornecimento de software e consultoria especializada para analisar de forma inteligente grandes quantidades de dados de uma organização. O uso de dados reais prolongou a fase de refinamento de dados, mas garantiu que os algoritmos e modelos desenvolvidos podem ser aplicado a situações reais. O grande conjunto de dados disponível abre um novo conjunto de possibilidades, e faz com que seja possível obter resultados interessantes e inovadores. Os modelos foram avaliados de acordo com três critérios: áres sob curva, sensibilidade e especificidade, com peso especial para os primeiros dois valores. O algoritmo random forest provou ser o mais adequado em todos os casos de teste.

Keywords: Previsão de churn; Telecomunicações; Machine Learning; Análise de churn; Churn clientes.

Acknowledgements

I would like to thank my supervisor, Professor João Moreira, for his work and invaluable advice. I would also like to thank all of my friends who have motivated me and taught me over the past years. I thank Rui for his support during the last years. Without him by my side, i would not have reached this far. A special thanks to my parents and family. Your effort made me accomplish this goal.

Georgina Cunha Esteves

"It's kind of fun to do the impossible."

Walt Disney

Contents

List of Figures

LIST OF FIGURES

List of Tables

Abbreviations

ANN	Artificial Neural Network
AUC	Area Under Curve
ARD	Automatic Relevance Determination
FEUP	Faculdade de Engenharia da Universidade do Porto
FN	False Negatives
FP	False Positives
HPC	High Performance Computing
IDE	Integrated Development Environment
KNN	K-nearest neighbors
MAE	Mean Absolute Error
PCC	Percentage of correctly classified
RMSE	Root Mean Square Error
ROC	Receiver operating characteristic
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machine
TP	True Positives
TN	True Negatives

Chapter 1

Introduction

Since the 1990s the telecommunications sector became one of the key areas to the development of industrialized nations. The main boosters were technical progress, the increasing number of operators and the arrival of competition. This importance has been accompanied by an increase in published studies sector and more specifically on marketing strategies [Gerpott et al., 2001].

In order to acquire new costumers, a company must invest significant resources to provide a product or service that stands out from the competitors, however, continuously evolving the product itself is not enough. Many companies are embracing new strategies based on data mining techniques to survive in this ever-increasing competitive market. These techniques are addressing challenging problems such as prospect profiling, fraud detection, and churn prediction. Churn refers to "the costumer movement from one provider to another" [Bott, 2014]. The ability to predict this variable is a concern of a broad number of industries, but it is mainly being focused by telecommunication service providers. There are several factors that justify this interest by telecom companies: the vast offer of similar products by multiple companies, along with the ease in changing operators and the Mobile Number Portability allow costumers to switch to another telecom provider effortless and still maintain their telephone number.

Companies were very fond of discovering the correlation between this new development and their profit, so they conducted a few studies [Wei and Chiu, 2002, Ascarza et al., 2016]. Three main strategies to generate more revenue were identified [Wei and Chiu, 2002]: acquire new customers, upsell the existing ones or increase customer retention. All these strategies will be compared using the return on investment value (ratio between extra revenue that results from these efforts and their cost). Gaining new customers is a well-known strategy in all markets to increase income. If a company is able to sell a product or service to a new customer, the profits of the company increase. However, this is considered to be an extremely expensive approach. The investment made regarding money, time and effort (new account setup, credit searches, and advertising and promotional expenses) is not so appealing when compared with the revenue produced. The second approach is upsell products to existing clients. This is considered to be a cheap strategy: the client

is already associated with the company, and he simply expands his current service or product to a more expensive one. But as seen in recent studies [Zaki and Meira, 2013], it is not an easy task to convince a customer to upgrade their current service. A big gain to the client must be present for them to consider this change. The strategy that was described as the most profitable (greatest return on investment) was increasing customer retention [Wei and Chiu, 2002]. The company profits in two ways when retaining a customer: they continue to generate revenue to the company by purchasing their service, and the competitors company does not gain strength in the market by gaining a new customer.

There are a vast number of strategies to improve customer retention overall. By simply talking to customers and conduct customer satisfaction surveys, a lot of problems regarding the customer-company relationship can be identified. But when the company wants to individualize customer retention, a new problem appears: a company has a great number of clients, and can not afford to spend a lot of time with each one of them. This is where Data Mining becomes useful: by predicting in advance when and which customers are at risk of leaving the company, all the efforts to retain customers could be addressed to these cases.

1.1 Context

Telecom companies are starting to realise the effectiveness of churn prediction as a way of generating more profit, specially when compared to other approaches. An increasing investment was made in the study area of churn prediction during past years [Ascarza et al., 2016]. In the context of a collaboration between FEUP and WeDo Technologies, an exploration of methods to detect which clients are close to change their telecom operator (churning) was conducted. WeDo technologies is the number one provider of revenue assurance and fraud management software for telecom operators [WeDoTechnologies,]. This area is of great interest for this kind of companies, because the costs of maintaining a current customer are typically lower than the costs of acquiring a new client[Reinartz and Kumar, 2002]. Churn prediction must be interpreted as a different problem for each company, because each business area has different indicators that must be identified to detect how attached a client is to the current operator.

1.2 Motivation and Goals

The objective of this dissertation is to develop a method for churn prediction in the telecom business. The intended result is an algorithm that identifies the clients that are closer to abandon their current operator. Although many approaches were conducted in the past few years, there are still many opportunities to improve the current work in this area. The data that will be used during this work is fundamental to assure the quality of the final solution. The dataset is an extend collection of calls conducted by real customers. This will extend the refinement time necessary, but will ensure that the developed algorithm and model can be applied to real world situations. It opens a new set of possibilities, and makes it possible to obtain interesting and novel results.

There are several desired requirements in a churn prediction system. According to the author [Balle et al., 2013], those requirements are the following:

- **Precision and Recall** - high level of recall (almost all churners must be identified) and medium-high level of precision (low number of false positives).
- **Performance** - the execution speed of the model with new data. This is essential for the company to be able to tale the right decisions at the right time.
- **Flexibility** - the model must be able to keep up with good forecast rates with new data. It should take into account changes in customers patterns when making the predictions.
- **Scalability** - the model needs to react positively if the data intake increases.
- **Targeting**: this feature relates to the ability to identify concrete data on the users more likely to leave the service.

All the requirements previously mentioned are taken into account when developing the solution to this problem. The most intricate requirement to achieve are the ones regarding the execution speed and model performance. Due to the big amount of available data to develop our solution, some of the tested algorithms take a considerable amount of time to complete. To neutralize this negative situation, we adopted a strategy based on parallel computation. In this way, the processing is distributed among the available cores, splitting the processing time by the number of cores (in the optimal scenario).

1.3 Dissertation Structure

Besides this introductory chapter, this dissertation contains 3 more chapters. In chapter 2, a review of the existing literature on the topic is conducted and some related work is presented. Several algorithms are detailed and explained, and data preprocessing importance and techniques are compared.

In chapter 3, an initial view around implementation details is made. An overview of the dataset and its variables is conducted, as well as the main operations made upon the data.

Chapter 4 contains the experimental setup definition and final results acquired.

Lastly, chapter 5 contains the main conclusions of this research, some ideas regarding future work.

Chapter 2

Literature Review on Churn Prediction

In this chapter a review of the bibliographic content found is conducted. The main focus are the approaches and techniques applied to churn prediction in telecom businesses, as well as some data preprocessing techniques and problems.

2.1 Customer Churn

According to the author [Yen and Wang, 2006], 'costumer churn' in telecom business refers to the costumer movement from one provider to another. 'Customer management' is the process conducted by a telecom company to retain profitable costumers.

The continuous evolution of technology has opened up the telecommunications industry, making this market more competitive than ever. These companies are realizing that a customer-oriented business strategy is crucial for sustaining their profit and preserving their competitive edge [Tsai and Lu, 2009]. As acquiring a new costumer can add up to several times the cost of efforts that might enable the firms to retain a customer, a best core marketing strategy has been followed by most in the telecom market: retain existing customers, avoiding customer churn [Kim and Yoon, 2004, Kim et al., 2004].

Two main types of targeted approaches to manage customer churn [Tsai and Lu, 2009] were identified: reactive and proactive. In the reactive approach, the company waits until the customer asks to cease their contract to act. In this situation, the company will then offer some advantages and incentives to retain the customer. The other approach is the proactive approach, where a company tries to identify which customers are more likely to churn before they do so. In this case, the company provides special offers to keep them from churning.

A few studies have been conducted to evaluate which is the approach, reactive or proactive, is better to a company. A majority of them agreed that the proactive approach achieves better results [Retana et al., 2016, Olaleke et al., 2014], and the work developed in this project will be based in

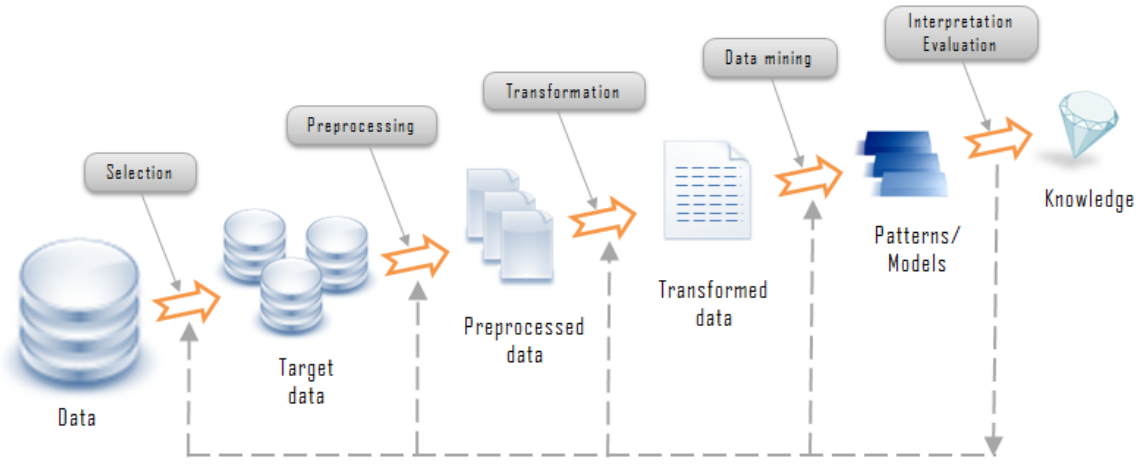
this second approach. The main objective is to predict beforehand the customers more likely to churn using data mining techniques.

In the scope of this project, a distinction between types of churn must be made. Involuntary churn is when circumstances outside the user and service provider's control affect the decision of ceasing their relationship. Customers' relocation to a distant location and death are part of this kind of churn. Voluntary churn is when a customer actively decides to leave the product or service of a company. Involuntary churn tends to be discarded to churn prediction, because those are the ones that do not represent the company-customer relationship.

2.2 Data Mining

The author [Yen and Wang, 2006] stated that data mining can be defined as 'the extraction of hidden predictive information from large databases'. It can be defined as a logical process that is used to search through large amount of data with the objective of finding useful information. The main goal of this technique is to find patterns that were previously unknown as well as novel information. This new information can be used by company owners to make adequate decisions to the companies future.

Figure 2.1: Data mining process



In Figure 2.1 a typical data mining process is described. That process can be simplified into three major steps: exploration, pattern identification and deployment. The first step is mainly focused on the data. It must be explored, cleaned and transformed (if necessary) to other forms of representation. Non-interesting data can also be discarded during this phase.

Once the data is explored and refined, a pattern identification must be formed. These patterns must be identified and chosen based on their performance on the prediction.

The last step regards the deployment of the patterns previously mentioned. The outcome must take into consideration the initial desire.

Data mining is, in fact, a new technology that has the potential to help the process of exploration of vast quantities of data.

When dealing with data from a big telecom company, only depending on human preprocessing effort is not efficient. With millions of clients and an ever bigger magnitude in the number of telephone calls, the process of scanning through all this information with only human intervention is both expensive and inefficient [Ngai et al., 2009]. The emerging data mining tools can answer this kind of problems. If the dataset has a sufficient size and quality, data mining technology can provide business intelligence to generate new opportunities [Yen and Wang, 2006].

2.3 Predictive modeling

In data mining, predictive modeling is the process of creating a model to predict an outcome. To do so, the past must be analyzed and, with this information, the model must be constructed and validated. In order to create this model there must be predictors, which are variable factors that expected to influence future behavior. For example, when predicting churn these factors can be number of calls to the operator assistance or the number of dropped calls regarding a certain client. If the outcome is qualitative it is called classification and if the outcome is quantitative it is named regression. Descriptive modeling or clustering is the assignment of observations into clusters so that observations in the same cluster are similar. Finally, association rules can find interesting associations amongst observations. The following sections will describe these techniques in detail, and identify specific types for each one.

2.3.1 Regression

Regression is a data mining function that predicts a numeric value. It can be used to model the relationship between one or more independent variables and dependent variables[Zaki and Meira, 2013]. For example: it could be used to predict children's height given their age and weight. In order to apply regression to a dataset, the target values must be known. Using the same example as above: to predict children's height, there must exist data during a period of time regarding their age, height, weight, and others. Height is then considered the target variable, and all the other attributes would be predictors.

During the model training process, a regression algorithm determines the value of the target. This value is assessed as a function of the predictors. The multiple iterations of this step threw all of the initial data compose a model. Later, this model can be applied to a different dataset with the unknown target values.

To better understand how regression works, a few basic concepts will be detailed. As already mentioned, the goal of regression analysis is to find the values of parameters in which the function best fits a set of provided data. The following equation represents this approach. The value of the continuous target variable (y) is the value of a function F according to predictors(x) and a set of patameters p. An error measure e is also taken into consideration, to minimize the possibility of overfitting.

$$y = F(x, p) + e \tag{2.1}$$

Four main groups of Regression algorithms are:

1. Frequency Table
    - Decision Tree
2. Covariance Matrix
    - Multiple Linear Regression
3. Similarity Functions
    - K Nearest Neighbors
4. Others
    - Artificial Neural Network
    - Support Vector Machine

The testing of regression models involves calculating multiple statistics. These statistics represent the difference between the predicted and the expected values.

Two of the most applied measures when testing a regression model [Shao and Deng, 2016] are Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

The MAE is an average of the absolute errors. This value is given by

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| \tag{2.2}$$

where *fi* is the predicted value and *yi* is the true value.

RMSE is the square root of the mean of the square of all of the error. The formula for RMSE is

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.3}$$

The use of RMSE is very common and it makes an excellent general purpose error metric for numerical predictions.

Compared to the previously detailed MAE, RMSE amplifies and punishes large errors.

2.3.2 Classification

Classification is a data mining task of predicting the value of a categorical variable (target or class) by building a model based on one or more numerical and/or categorical variables (predictors or attributes) [Zaki and Wong, 2003]. The objective is to predict the most accurate outcome based on

a set of inputs. In order to predict this, the algorithm processes a training set containing several attributes and their respective outcome (prediction attribute). It then tries to discover relationships between these attributes that lead to predicting the outcome. Next, a new set must be taken into consideration: the prediction set. This is composed of new data (unknown to the algorithm) with the same attributes as the training set except for the prediction attribute. The algorithm produces a prediction based on this new data, and an accuracy value defines how "good" the algorithm is. Applications regarding fraud detection and credit-risk are normally connected to this technique.

Classification algorithms can be stratified into four main groups:

1. Frequency Table
   - ZeroR
   - OneR
   - Naive Bayesian
   - Decision Tree
2. Covariance Matrix
   - Linear Discriminant Analysis
   - Logistic Regression
3. Similarity Functions
   - K Nearest Neighbors
4. Others
   - Artificial Neural Network
   - Support Vector Machine

Table 2.1: Confusion matrix

		Predicted	
		Non churn	Churn
Actual	Positive	TP	FN
	Negative	FP	TN

In order to assess the developed models, evaluation methods must be applied. The confusion matrix showed in table 2.1 contains information about actual and predicted classifications done by a classification system.

Each term corresponds to a specific case:

- **True positives (TP):** These are cases in which the prediction is yes (the client did not left the company), and he did stay.

- **True negatives (TN):** The prediction is no, and they churned (left the company).
- **False positives (FP):** The prediction is yes, but the client left the company. (Also known as a "Type I error.")
- **False negatives (FN):** The prediction is no, but the client actually stayed in the company. (Also known as a "Type II error.")

The two types of errors the model can commit, FP and FN, will also have different weights in the system. It is worse to a telecom company not to predict a customer that is going to churn, than to address one that does not plan to leave the company.

Three criteria will be used to evaluate the system: accuracy, hit rate and churn rate.

Accuracy measures the rate of the correctly classified instances of both classes. The formula is:

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \tag{2.4}$$

Hit rate gives us an index of the rate of predicted churn in actual churn and actual non-churn. It is represented by:

$$HitRate = \frac{TN}{FN+TN} \tag{2.5}$$

Churn rate measures the rate of predicted churn in actual churn. The formula is:

$$ChurnRate = \frac{TN}{FP+TN} \tag{2.6}$$

2.3.2.1 K Nearest Neighbors

K Nearest Neighbors (KNN) is a simple algorithm that collects all available cases and classifies new cases based on a similarity measure. KNN has been used in statistical estimation and pattern recognition since the beginning of 1970's as a non-parametric technique.

The process of assigning a class to a new case follow these steps:

1. Find k nearest neighbors of the new case in the existing dataset, according to some distance or similarity measure. It compares the new sample to all known samples in the existing dataset and determines which k known samples are most similar to it. K value is determined beforehand.
2. Determine which class value is the one most of those k known samples belong.
3. Assign the determined class to the new sample.

If the target variable is categorical, there are three most used distance measures: Euclidean, Manhattan and Minkowski.

In the instance of categorical variables the Hamming distance must be used. This measure is calculated according to the following equation.

$$D_H = \sum_{i=1}^{k} |x_i - y_i| \tag{2.7}$$

2.3.2.2 Naive Bayes

The Naive Bayesian classifier is based on Bayes' theorem. It is a model easy to build, with no complicated iterative parameter estimation, which makes it useful and practical for very large datasets. Despite its simplicity, the Naive Bayes classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

Bayes theorem provides a way of calculating the posterior probability P(c|x). It assumes that the effect of the value of a predictor (x) on a given class (c) is independent of the values of the other predictors. This assumption is called class conditional independence.

The posterior probability is calculated according to the following equation:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \tag{2.8}$$

Where:

- P(c | x) is the posterior probability of the target given predictor
- P(x | c) is the probability of predictor given the target
- P(c) is the probability of target
- P(x) is the probability of predictor

2.3.2.3 C4.5

C4.5 is an algorithm used to generate decision trees from a set of training data. Each node in the tree corresponds to a non-categorical attribute and each arc to a possible value of that attribute. A leaf of the tree specifies the expected value of the categorical attribute for the records described by the path from the root to that leaf.

At each node of the tree, the algorithm chooses the attribute of the data that most effectively splits its set of samples into subsets. The splitting criterion is the normalized information gain. The attribute with the highest normalized information gain is chosen to make the decision upon. Usually the category attribute takes only the values true, false, or something similar where one of its values represents failure.

Regarding the algorithms behavior, we can describe 3 base cases:

- All the samples belong to the same class; in this case, the algorithm creates a leaf node saying to choose that class.
- None of the analyzed features provides information gain; in this case, C4.5 creates a node higher up on the decision tree with the expected value of the class.
- Finds an instance of previously unseen class; the algorithm behaves the same way as in the previous point.

2.3.2.4 Random Forest

Random Forest is a technique that grows many classification trees. The growth of each tree follows the steps:

1. Considering N the number of cases in the training set, sample N cases at random from the original data to grow the trees.
2. Considering M as the number of input variables, a number m lower than M is specified. At each node, m variables are selected and the best split on these m is used to split the node. Value m is not altered during the trees growth.
3. Each tree is growth to largest extent possible without pruning the trees.

To classify a new individual, that individual is tested with all the previously generated trees. Each tree gives a classification, and the classification that acquired the most number of votes is considered to be correct for that individual.

2.3.2.5 AdaBoost

AdaBoost is a type of ensemble learning algorithm where multiple learners are employed to build a stronger learning algorithm. It works by choosing a base algorithm (e.g. decision trees) and iteratively improving it by accounting for the incorrectly classified examples in the training set.

It operates in the following way:

1. Assign equal weights to all training examples and chose a base algorithm
2. At each step of iteration, apply the base algorithm to the training set and increase the weights of the incorrectly classified examples
3. Iterate n times, each time applying base learner on the training set with updated weights
4. The final model is the weighted sum of the n learners

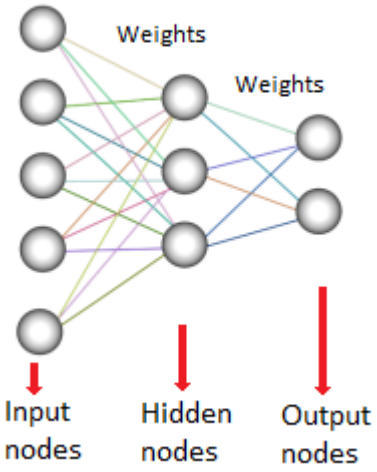
2.3.2.6 ANN

Artificial Neural networks can be interpreted as a brain metaphor for information processing. The ability to learn from the data and to generalize have popularized this method. According to the author [Tsai and Lu, 2009], neural computing refers to a pattern recognition methodology for machine learning. The resulting model from neural computing is called artificial neural network (ANN). They are mostly applied in business applications for pattern recognition, forecasting, prediction, and classification.

Figure 2.2 represents the network structure of a ANN. It is composed of multiple neurons grouped in three layers: input, hidden and output.

The input layer corresponds to the values of the set of attributes from the dataset.

Figure 2.2: Neural Network



The hidden layer takes the inputs from the previous layer and converts them into outputs for further processing. This layer can be interpreted as a feature extraction mechanism.

The output layer contain the solution to a problem.

There can be multiple hidden layer in a ANN, but often just one is used.

The key element in a ANN are connection weights. They represent the relative importance of each input to a processing element. These weights are continuously adjusted, which allows the learning process of the ANN.

2.4 Churn Prediction Approaches

The ability to predict that a particular customer has a high risk of churning, while there is still time to do something about it, represents a huge additional potential revenue source for every business. So, a huge investment has been conducted in this area, and multiple approaches have been studied and tested.

The accuracy of the techniques used is critical to the success of any retention efforts. In the following sections, some of these technique applied to the churn problem will be presented.

2.4.1 Decision tree

Most customer segmentation methods are based on experience or Average Revenue per User, and do not take into consideration customers' future revenue or the cost of servicing customers of different types. A costumer evaluation by these methods may not be the most accurate. In this approach [Han et al., 2012], a new way of customer segmentation is proposed. This model can be used to predict a customer lifecycle using only his demographic information. For each customer,

five decision models are applied: current value, historic value, prediction of long-term value, credit and loyalty. To compute loyalty and credit, an AHP (analytic hierarchy process) method is used.

For evaluating the accuracy of the model, the hit ratio of customer value is taken into consideration.

2.4.2 Logistic Regression

This study [Oghojafor et al., 2012] uses logistic regression to examine the effect of socio-economic factors on customer attrition. This is done by investigating the factors that influence subscribers churning one service provider for another. In this study, "Intention to drop current service provider" is taken as the categorical response variable. The objective is to evaluate the impact of some demographic and socio-economic factors on the willingness to churn. These factors are age, sex, marital status, education, income, occupational type, occupation and advertising medium.

Two models were constructed in this research. The first one used the most common factors in churn prediction to construct the model, such as call expenses, type of service and number of mobile connection. The second model used the factors from the first model and a few more: the demographic and socio-economic factors.

This approach concluded that most independent variables are highly significant to the prediction, wich indicates a "strong relationship between the independent variables and the explanatory variable". High call rates and poor service facilities stand out among the identified churn determinants.

2.4.3 Neural Network

Hybrid data mining techniques combining two or more techniques have been gaining visibility. In a vast number of problem domains, these techniques are proving to provide better performances than single techniques.

One of the studied approaches [Tsai and Lu, 2009] considers two hybrid models. It combines two different neural network techniques for churn prediction: back-propagation artificial neural networks (ANN) and self- organizing maps (SOM). The hybrid models are ANN combined with ANN and SOM combined with ANN. The first technique of the two models performs data reduction task by filtering out unrepresentative training data. Then, the outputs are used to create the prediction model based on the second technique. In order to evaluate the performance of these models, three different kinds of testing sets were developed. The experimental results showed that the two hybrid models outperform the single neural network baseline model in terms of prediction accuracy and Types I and II errors over the three kinds of testing sets.

Another study [Bott, 2014] investigates the application of Multilayer Perceptron (MLP) neural networks with back-propagation learning to identify the most influencing factors in costumer churn. Two methods were examined and compared; the typical change on error method and the ANN weights based method. In both methods, three attributes were identified as important to the

prediction by both models. These attributes are total monthly fees, total of international outgoing calls and 3G service.

2.4.4 Support vector machines

One of the studied approaches[Coussement and Poel, 2008] consisted on applying SVM to a newspaper subscription churn context. The objective was to construct an accurate churn model using and tunning this technique. The customer churn prediction performance of the model was benchmarked to logistic regression and random forecasts.

The authors chose as the main metric to evaluate their models Area under curve (AUC). The best value regarding their SVM models was 85.14, and the random forest model used as benchmark conquered a final AUC value of 87.21.

2.4.5 Multiple methods

According to our research literature regarding customer churn, the majority of the related work focuses on applying only one data mining method to extract knowledge.

Only a few authors [Verbeke et al., 2012] focused on comparing multiple strategies to predict customer churn. One of the investigated studies [Buckinx and den Poel, 2005] consists on comparing three classification techniques: Logistic regression, automatic relevance determination (ARD) Neural Networks and Random Forests. The models were evaluated regarding their AUC and percentage of correctly classified (PCC) instances, in both training and testing sets.

The highest AUC value was obtained by the Random Forest model in both train and test sets, with values 0.8249 and 0.8319 respectively.

2.5 Data Preprocessing

Data preprocessing is an important step on any data mining project. Real world data is generally incomplete, inconsistent and even with errors. In order to produce a good model to a problem, these issues must be addressed. It will be discussed several data preprocessing issues and techniques.

2.5.1 Data Problems

This section classifies the major data quality problems to be solved by data cleaning and data transformation. As we will see, these problems are closely related and should thus be treated in a uniform way. Cleaning a target dataset can be an even heavier task then collecting the data [Rahm and Do, 2000]. Data can contain several kinds of problems:

- Noise : The occurrence of noise in data is typically due to recording errors and technology limitations. In can also be connected to the uncertainty and probabilistic nature of specific feature and class values.

- Missing data : Missing data can occur due to multiple situations. There could be conflicts in the recorded data, which lead to overwritten data. In some cases, that specific field of the data could not be considered important at the time and hence not captured.
- Redundant data : This kind of error is mostly related with human errors. The data could have been recorded under different names or in different places. It can also be representative of records containing irrelevant or information-poor attributes.
- Insufficient and stale data : Sometimes the data that we need comes from rare events and hence we may have insufficient data. Sometimes the data may not be up to date and hence we may need to discard it and may end up with insufficient data.

2.5.2 Data Preprocessing techniques

In order to improve the quality of the previously gathered data, several preprocessing techniques can be applied [Zaki and Meira, 2013]. These techniques can be divided into four major categories: data cleaning, data transformation, data reduction and data integration.

2.5.2.1 Data cleaning

Data cleaning techniques aim to clean the data by filling in missing values, dealing with outliers, smoothing noisy data and fixing inconsistencies. When the data contains missing values, a few methods can be implemented to fix this issue. The most common approaches to solve this problem are ignoring the tuple, or fill in the missing value using the *mean* attribute.

If the data contains noise, some data smoothing techniques can be applied, including clustering and regression methods.

Inconsistent data may be corrected through a paper trace method. Knowledge engineering tools can also be used to detect violations of known data constraints.

2.5.2.2 Data transformation

Data transformation consist in reconstructing the data into a new form more appropriate to the data mining task. Several kinds of data transformation can be employed:

- Normalization
- Smoothing
- Data Generalization
- Aggregation

Normalization is one of the most used techniques to transform data. In the simplest case, it means adjusting values that are measured on different scales to a common scale. It many cases, this is conducted after averaging the numbers.

2.5.2.3 Data reduction

Mining on big amounts of data can be a longstanding task, sometimes even unfeasible. Data reduction techniques aim to reduce the quantity of data to be analyzed without compromising the integrity of the original data. It consists on reducing the volume of the data or its dimensionality, by removing attributes. There are a few strategies for data reductions:

- Data cube aggregation
- Data compression
- Data reduction
- Numerosity reduction
- Discretization

2.5.2.4 Data integration

Most of data analysis projects involve combining data from multiple source into a single data store. This task is called data integration. Several challenges are presented when integrating multiple data sources. For example, the same attribute can have different names among the different sources, and it may not have an intuitive name.

The best approach in this kind of task is to use the metadata usually present in databases and warehouses to help avoid error in the integration.

2.6 Conclusion

Several predictive modeling approaches were studied and their main characteristics detailed. Besides this overview, a more detailed explanation of some authors approaches to predict churn is presented. Most of the authors focus on training and testing one single model, tunning its parameters to best fit their main objective.

We considered that an approach consisting of testing multiple algorithms and compare their results would bring interesting results to our research. With this mindset, we selected six algorithms from multiple backgrounds. Those algorithms are:

- KNN
- Naive Bayes
- Random Forest
- C4.5 (J48 implementation)
- AdaBoost

- ANN

The choice of these algorithms took into consideration their popularity and efficiency among the data mining community, as well as some studies that compare data mining approaches [Wu et al., 2008, Fern and Cernadas, 2014] in classification problems.

Chapter 3 will focus the dataset itself, its analysis and some of the preprocessing techniques applied.

Chapter 3

Dataset

The models that predict customer churn are based on knowledge regarding the company's clients and their calls. That information is stored in a database table and is called dataset.

All the models were trained and tested with this data. But before that can happen, this collection had to go through multiple transformations and pre-processing techniques to make it suitable to predict upon.

This chapter aims to describe all the steps regarding data. It begins explaining how the data was retrieved and stored, how we conducted the selection process of the needed information, its analysis and the main transformations performed to make it suitable to predict upon.

3.1 Data collection

For the purpose of this research, real data from WeDo technologies client calls was provided by the company. The data is stored in a SQL file, and has a size of more than 131 gigabytes. The file contains one table with over 1.2 billion entries from 5 million different clients.

SQLite library[SQLite,] was chosen due to its speed and efficiency, and since it is the technology adopted by the High Performance Computing (HPC) [FEUP,] at Faculdade de Engenharia Universidade do Porto (FEUP) . HPC is a research tool that can be used to solve complex computational problems. This technology added great value to our development process, because it provided great computational power that enabled faster processing times.

3.2 Data selection

Due to computational and time limitations, the dataset available to this research was too large and needed to be sampled. In order to prevent the lost of valuable information and keep the final results accurate, a simple selection of the first N table entries was not viable. The right approach to this problem is to retrieve all the call records from a group of clients. The query employed to divide the database was the following:

```
SELECT * FROM call_details AS c INNER JOIN
(SELECT DISTINCT(contract_id) FROM (
SELECT * FROM call_details LIMIT 1000000))
 AS v ON c.contract_id = v.contract_id;
```

In this way, we the dataset consists of entire history of a percentage of the total number of clients to predict upon. The final dataset generated by this query contained over 100 thousand calls between 30 June 2012 and 31 January 2013.

3.3 Data analysis

The dataset available to this research contains information regarding call information of telecom company clients.

Table 3.1: Variables in the data

Variable	Value	Description
DATE	YearMonthDay	Date of the call
TIME	HoursMinutesSeconds	Time of the call
DURATION	Seconds	Call duration
MSISDN	Numeric	Anonymized number. If Incoming, it is the number getting the call. If Outgoing, is the number calling.
OTHER_MSISDN	Numeric	The "other" number in the call (regarding the previous variable).
CONTRACT_ID	Numeric	Client code
OTHER_CONTRACT_ID	Numeric	Not present in all data (does not belong to the operator).
START_CELL_ID	Numeric	Should represent the calling device (Outgoing).
END_CELL_ID	Numeric	Should represent the device getting the call (Ingoing).
DIRECTION	I, O	Represent a incoming call,("I") or outgoing,("O").
CALL_TYPE	FI, MO, ON, OT, SV	FI - The other device in the call bellongs to a wireline MO - The other device in the call bellongs to amobile network ON - On-Net, both incoming and outgoing systems bellong to this operator OT - Others SV - Services:VoiceMail calls, etc
DESTINATION_TYPE	I, L	Local (L) or international (I) desteny.
DROPPED_CALL	Y, N	Dropped call (Yes or No)
VOICMAIL	Y, N	Call went to voicemail (Yes or No)

Figure 3.1: Data structure

```
> str(dados)
'data.frame':   100064 obs. of  13 variables:
 $ timestamp        : int  1341133745 1341133779 1341134477 1341135678 1341142822 1341142899 134114
9426 1341218098 1341218152 ...
 $ duration         : int  29 16 157 134 61 11 83 83 19 36 ...
 $ msisdn           : int  24832136 24832136 24832136 24832136 24832136 24832136 24832136 24832136
832136 ...
 $ other_msisdin    : int  17879828 17879828 17879828 23744915 17879828 17879828 17879828 17879828
879828 ...
 $ contract_id      : int  4261599 4261599 4261599 4261599 4261599 4261599 4261599 4261599 4261599
 $ other_contract_id: int  -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
 $ start_cell_id    : int  1762 1762 3728 0 0 3839 3839 4870 3002 3002 ...
 $ end_cell_id      : int  0 0 0 2676 3839 0 0 0 0 0 ...
 $ direction        : chr  "O" "O" "O" "I" ...
 $ call_type        : chr  "MO" "MO" "MO" "MO" ...
 $ destination_type : chr  "L" "L" "L" "L" ...
 $ dropped_call     : chr  "N" "N" "N" "N" ...
 $ voicemail        : chr  "N" "N" "N" "N" ...
```

Table 3.1 contains an overview of the data. It identifies the variables, their values and a simple description of their meaning.

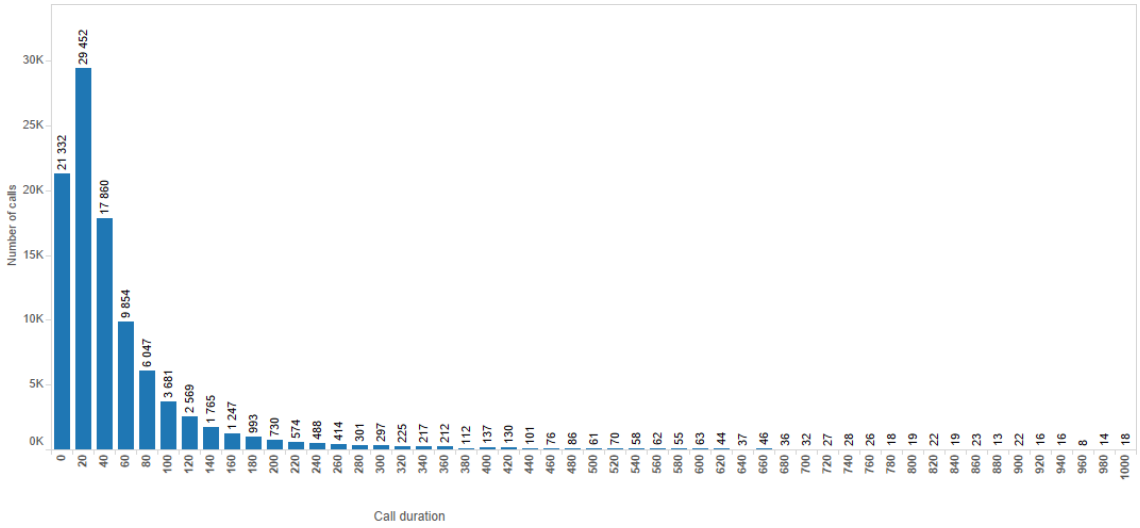
To explore data and also to implement and evaluate the algorithms we chose R language and environment [Foundation,]. This data was stored in a data frame, and contains categorical (character) and continuous (numeric) values. A variety of useful functions to explore the data frame were applied, namely *str()* and *summary()* functions.

The *str()* function returns a compact display of the internal structure of the data frame. It returns a few example outputs and the types of data for each column. As represented in 3.1, there are two types of variables: *int*, which are the integer variables; and *chr*, representing the character variables.

Figure 3.2: Data summary

```
> summary(dados)
   timestamp            duration           msisdn         other_msisdin       contract_id
 Min.   :1.341e+09   Min.   :   1.00   Min.   :  244575   Min.   :     563   Min.   :  14940
 1st Qu.:1.343e+09   1st Qu.:  22.00   1st Qu.: 7651151   1st Qu.: 7657046   1st Qu.:1606869
 Median :1.353e+09   Median :  39.00   Median :19183377   Median :15232724   Median :2580053
 Mean   :1.350e+09   Mean   :  69.03   Mean   :16514724   Mean   :15391658   Mean   :2702057
 3rd Qu.:1.356e+09   3rd Qu.:  71.00   3rd Qu.:22878255   3rd Qu.:22993025   3rd Qu.:4017427
 Max.   :1.360e+09   Max.   :7132.00   Max.   :31232100   Max.   :31453805   Max.   :5263806
 other_contract_id start_cell_id   end_cell_id      direction          call_type
 Min.   :     -1   Min.   :   0   Min.   :   0.0   Length:100064      Length:100064
 1st Qu.:     -1   1st Qu.:   0   1st Qu.:   0.0   Class :character   Class :character
 Median :     -1   Median :1140   Median :   0.0   Mode  :character   Mode  :character
 Mean   : 961484   Mean   :1159   Mean   : 977.3
 3rd Qu.:1777401   3rd Qu.:1871   3rd Qu.:1589.0
 Max.   :5115514   Max.   :5275   Max.   :5285.0
 destination_type   dropped_call       voicemail
 Length:100064      Length:100064      Length:100064
 Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character
```

Figure 3.3: Number of calls per call duration

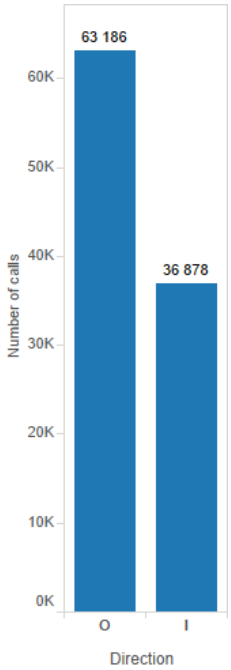


summary() is a generic function used to produce result summaries of data. When applied to a data frame, it is applied to each column, and the results for all columns are shown together. The results of this function applied to the data are shown in 3.2. As the values were not yet transformed into proper data types, some of the columns are not interpreted the right way, such as "timestamp". In the variable "duration", the maximum value is extremely high when compared to that variables' mean.

To acquire a better understanding of some of the data variables, we used Tableau Desktop[Software,] software. Tableau is a data visualization and communication tool widely used around the world. It can import in an easy and quick way data from multiple file formats, including R objects.

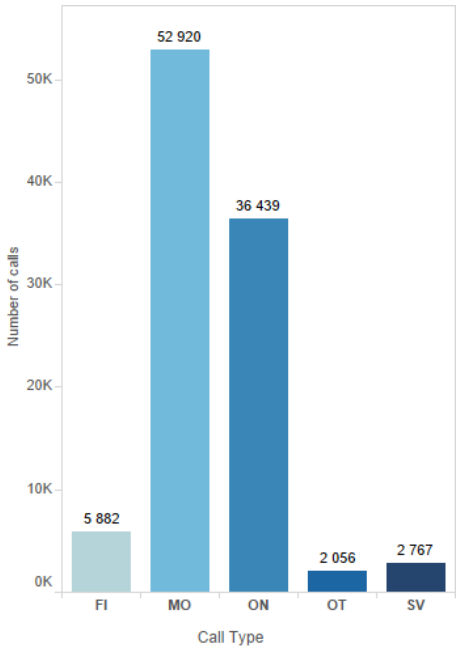
Figure 3.3 is a visual representation of the duration in minutes of client calls. The calls were group in 20 minutes intervals. The graph was also trimmed at 1000 minutes to make the graph more easily understandable. We can conclude that the majority of the calls on our dataset have a duration between 1 and 60 minutes. This number continues decreasing with the growth of the call duration, and stabilizes near 440 minutes.

Figure 3.4: Call direction



In our dataset, variable "direction" expresses if a call record regards an incoming call ("I") or an outgoing call ("O"). That variable can be further analyzed on figure 3.4. The majority of the calls represent an outgoing call, and only around 35 thousand are incoming calls.

Figure 3.5: Call type



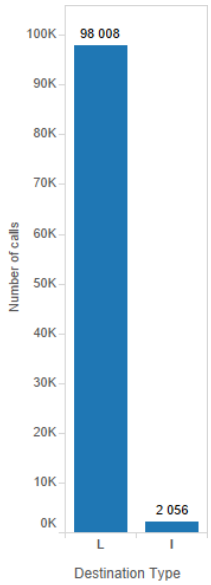
A call can be labeled in one of five groups:

- Fixed (FI) - the other person on that call belongs to a landline.
- Mobile (MO) - the other person on that call belongs to a mobile network.
- On-Net (ON) - both ends of the call belong to WeDo operator.
- Services (SV) - voicemail, premium-rate, etc.
- Others (OT) - all the calls that do not belong to the previously mentioned groups.

Figure 3.5 shows the values dispersion regarding the type of the call. Around 90% of the calls belong to the Mobile and On-Net groups.

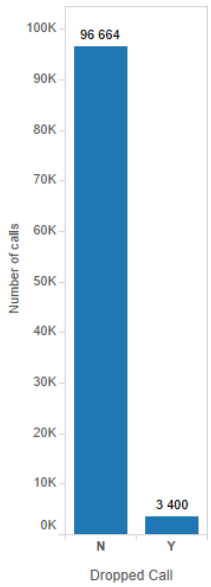
As shown in figure 3.6, the majority of the calls are between numbers from the same country, named local (L) calls. Only 3% of the calls in our dataset are international (I).

Figure 3.6: Call destination



A similar result can be found when analyzing dropped calls data. A call is said to be dropped when due to technical reasons, is cut off before the speaking parties had finished their conversation and before one of them had hung up. On 3.7 we can see that a minority of our data has this specific characteristic.

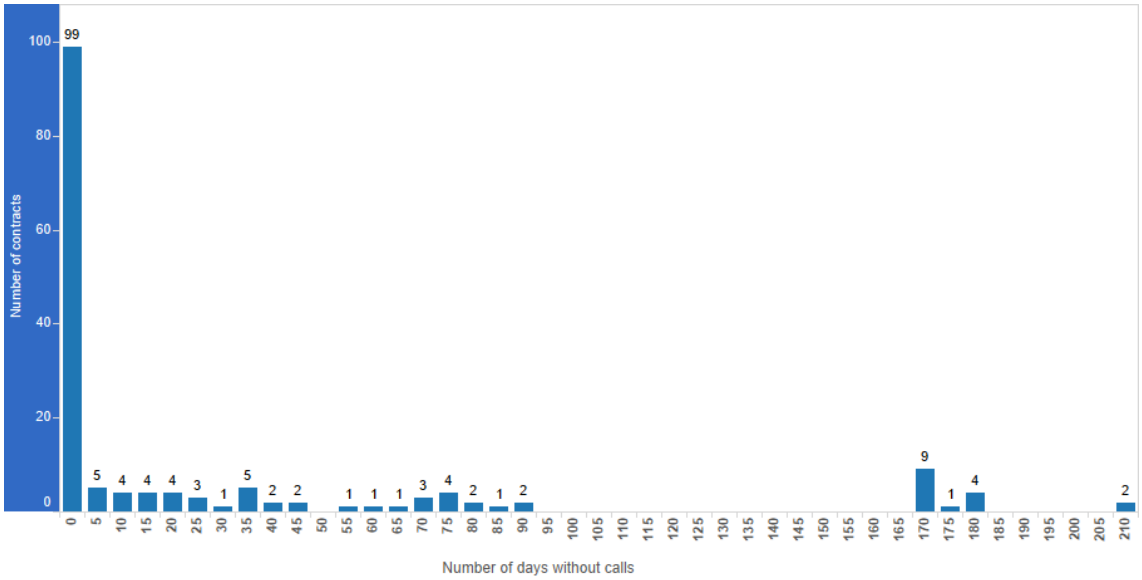
Figure 3.7: Dropped calls



The main problem regarding our data was the lack of knowledge of what costumers churned and the ones that did not. According to some authors [Oentaryo et al., 2012, Radosavljevik et al., 2010], we can state that a costumer has churned a telecom company when he does not do or receive any communication during 30 days or more.

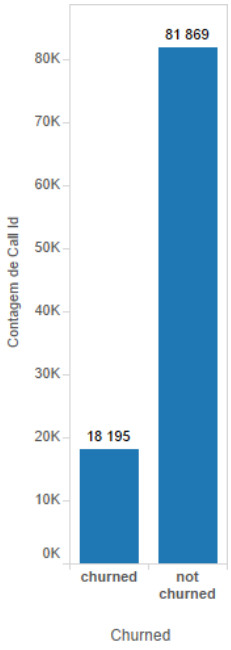
The respective calculations were conducted, and final results are shown in figure 3.8. More than 60 % of clients have done communications in less then 5 days, which tells they are currently active. We also have an estimate of approximately 26% customers that churned the company, with a maximum day difference of 210 days without communications.

Figure 3.8: Days without calls



The final distribution of the clients concerning churn is represented in figure 3.9. This is our target variable, which means that the objective of the developed models is to predict the outcome of this variable through the study of the other variables in our data. We can already state that our target class is imbalanced, which means that the classes are not represented equally. This problem will be addressed in the next chapter.

Figure 3.9: Churn distribution



3.4 Data transformation

Some algorithms in data mining require the dataset to have specific characteristics. The first step in data transformation was to convert data types to most adequate ones. In figure 3.10 is represented the new data structure after this conversion. Most of the variables were converted to numeric type to make the dataset easily adaptable to multiple algorithms. The exception in this conversion was the variable "churned" previously mentioned on section 3.3. These variables were defined as a factor. Factor variables are categorical variables that can be either numeric or string.

Figure 3.10: Data structure after type conversion

```
> str(new_data)
'data.frame':   100064 obs. of  16 variables:
 $ call_id          : num  1 2 3 4 5 6 7 8 9 10 ...
 $ churned          : chr  "churned" "churned" "churned" "churned" ...
 $ days_difference  : num  38.7 38.7 38.7 38.7 38.7 ...
 $ timestamp        : num  1.34e+09 1.34e+09 1.34e+09 1.34e+09 1.34e+09 ...
 $ duration         : num  29 16 157 134 61 11 83 83 19 36 ...
 $ msisdn           : num  24832136 24832136 24832136 24832136 24832136 ...
 $ other_msisdin    : num  17879828 17879828 17879828 23744915 17879828 ...
 $ contract_id      : num  4261599 4261599 4261599 4261599 4261599 ...
 $ other_contract_id: num  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
 $ start_cell_id    : num  1762 1762 3728 0 0 ...
 $ end_cell_id      : num  0 0 0 2676 3839 ...
 $ direction        : num  0 0 0 1 1 0 0 0 0 0 ...
 $ call_type        : num  1 1 1 1 1 1 1 1 1 1 ...
 $ destination_type : num  0 0 0 0 0 0 0 0 0 0 ...
 $ dropped_call     : num  0 0 0 0 0 0 0 0 0 0 ...
 $ voicemail        : num  0 0 0 0 0 0 0 0 0 0 ...
```

Figure 3.11: Summary of new data

```
> summary(new_data)
    call_id         churned          days_difference      timestamp           duration
 Min.   :     1   Length:100064      Min.   :  0.0000   Min.   :1.341e+09   Min.   :   1.00
 1st Qu.: 25017   Class :character   1st Qu.:  0.1104   1st Qu.:1.343e+09   1st Qu.:  22.00
 Median : 50033   Mode  :character   Median :  0.2277   Median :1.353e+09   Median :  39.00
 Mean   : 50033                      Mean   : 26.7724   Mean   :1.350e+09   Mean   :  69.03
 3rd Qu.: 75048                      3rd Qu.:  4.1121   3rd Qu.:1.356e+09   3rd Qu.:  71.00
 Max.   :100064                      Max.   :214.0824   Max.   :1.360e+09   Max.   :7132.00
     msisdn          other_msisdin       contract_id      other_contract_id start_cell_id   end_cell_id
 Min.   :  244575   Min.   :     563   Min.   : 14940   Min.   :     -1   Min.   :   0   Min.   :   0.0
 1st Qu.: 7651151   1st Qu.: 7657046   1st Qu.:1606869   1st Qu.:     -1   1st Qu.:   0   1st Qu.:   0.0
 Median :19183377   Median :15232724   Median :2580053   Median :     -1   Median :1140   Median :   0.0
 Mean   :16514724   Mean   :15391658   Mean   :2702057   Mean   : 961484   Mean   :1159   Mean   : 977.3
 3rd Qu.:22878255   3rd Qu.:22993025   3rd Qu.:4017427   3rd Qu.:1777401   3rd Qu.:1871   3rd Qu.:1589.0
 Max.   :31232100   Max.   :31453805   Max.   :5263806   Max.   :5115514   Max.   :5275   Max.   :5285.0
   direction        call_type     destination_type   dropped_call       voicemail
 Min.   :0.0000   Min.   :0.000   Min.   :0.00000   Min.   :0.00000   Min.   :0.000000
 1st Qu.:0.0000   1st Qu.:1.000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000000
 Median :0.0000   Median :1.000   Median :0.00000   Median :0.00000   Median :0.000000
 Mean   :0.3685   Mean   :1.429   Mean   :0.02055   Mean   :0.03398   Mean   :0.007795
 3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.000000
 Max.   :1.0000   Max.   :4.000   Max.   :1.00000   Max.   :1.00000   Max.   :1.000000
```

By the analysis of figure 3.11, we can clearly detect that there are big differences among the variable ranges. Values for duration feature range between 1-7132, and values for msisdn feature range from 244575-31232100. This kind of variation could impact prediction accuracy

[C.Saranya, 2013]. The objective is to improve predictive accuracy and not allow a particular feature to impact the prediction due to large numeric value range. Thus, we normalized the values. 2

Chapter 4

Implementation and Results

Regarding the models development, tunning and comparison, we chose R language [Foundation,] to implement and conclude the previously mentioned tasks. R is a language and environment for statistical computing and graphics. It provides a wide variety of statistical and graphical techniques, and is highly extensible. The R installation already provides a software environment, but we chose to use RStudio [R Core Team, 2013] as our integrated development environment (IDE). When compared with the built-in version, it provides additional features considered to be relevant to the improvement of the development process.

4.1 Experimental Setup

To guarantee the results integrity when comparing multiple algorithms, we need to assure that all the algorithms are tested in the same conditions and with the same data mentioned on section 3.4. As the original data was ordered, we applied a randomize function to the dataset. We also removed variables "contract_id" and "msisdn" from the data that was going to train the multiple models, because they have a direct connection with the outcome of target variable (identify the client itself).

The solution found consists in dividing the original data into two subsets: train set and test set with a 70/30 ratio.

The train set is formed by 70% of the original data, and aims to train the algorithm.

The test set contains 30% of the initial dataset, and is used to minimize overfitting and to tune the algorithms parameters.

Has we have already mentioned on the previous section, our target class is imbalanced, and this value disparity could have a significant negative impact on the final models regarding model fitting. Three main sampling methods were studied to solve this issue:

- Down-sampling
- Up-sampling
- Hybrid methods

The first method consists on randomly dividing all the classes items in the training set so that their frequencies match the least common class.

The act of up-sampling data consists on randomly sampling the class with the lowest frequency to be the same size as the majority class.

Hybrid methods can be considered middle ground of the previously mentioned techniques. This methods down-sample the majority class (churned) and synthesize new data points in the minority class (not churned).

The chosen approach was a hybrid sampling method [Chawla et al., 2002] named Synthetic Minority Over-sampling Technique (SMOTE) that was applied to the training set. This method approaches the data in two ways: generates new examples of the minority class using the nearest neighbors of that cases, and under-samples the majority class.

In table 4.1 are the results of the applied sampling method. The first row of that table corresponds to the dataset before the sampling method, and the second row to the data generated by SMOTE method.

Table 4.1: Dataset Sampling results

	Churn	**Churn %**	**Not Churn**	**Not Churn %**	**Total**
Original	18195	18,2	81869	81.8	100064
SMOTE	38211	42.9	50948	57.1	89159

The evaluation metrics chosen are Area Under Curve (AUC), sensitivity and specificity. The first approach was to use accuracy as a metric to evaluate the models, but due to the unbalancing target class, where there was a low percentage of samples in one of the classes as mentioned on figure 3.9, it was not good metric to be applied to our models. Next we present a brief description of the evaluation metrics used:

- Specificity: Also called *true negative rate*, corresponds to the number of churned cases correctly identified divided by the total number of negatives (all the churn cases). Essentially, specificity represents the percentage of correctly classified cases when a costumer has not churned.

- Sensitivity: Also known as *true positive rate*, measures the proportion between positive examples which were predicted as positive. In our case, sensitivity means the percentage of detected cases when a costumer has actually churned.

- AUC: A Receiver Operating Characteristic (ROC) chart is a curve representation of the proportion of false positives (1- specificity) on the horizontal axis against the proportion of true positives on the vertical axis (sensitivity). This graph can be used to determine the optimal balance between sensitivity and specificity. AUC is a measure used to compare accuracies of multiple classifiers and to evaluate how well a method classifies. The closer to 1 the AUC of a classifier is, the higher accuracy the method has.

In customer churn analysis it might be more expensive to incorrectly infer that customer is not churning then to give a general reduction in prices for services to clients that are not planning to leave the company. Since the priority in our case study is given to identifying churn clients rather than not churn ones, Sensitivity is more relevant than Specificity in our results.

The algorithms applied were chosen due to their diversity of representation and learning style, and their common application on this kind of problems. We also took into consideration studies regarding the popularity and efficiency [Wu et al., 2008].

Six different machine learning models were trained and compared among themselves. Those algorithms were:

- Knn
- Naive Bayes
- Random Forest
- C4.5
- AdaBoost
- Ann

All the models were trained using functions available on R's package *caret*[CRAN,], which contains "functions for training and plotting classification and regression models".

Each model was tuned and evaluated using 3 repeats of 10-fold cross validation, a common configuration on data mining for comparing different models [Kuhn, 2008]. The model with the best scores is then chosen to make predictions in new data, defined as test set.

A random number seed is defined before the train of each one of the algorithms to ensure that they all get the same data partitions and repeats.

The training of the models was done using the same data and control metrics, to establish a solid base for the future model comparisons. After the training phase, we made statistical statements about their individual results and performance differences. All the resampling results were collected to a list using *resamples()* function, which checks that the models are comparable and that they used the same training scheme (*trainControl* configuration). It also contains the evaluation metrics for each fold and each repeat per algorithm. The results and deep analysis of the algorithms results will be presented on the next section.

All the computation during this research was done with the resources at FEUP Grid [FEUP,], and using a library for parallel computation [Weston and Calaway, 2015].

4.2 Results

In order to acquire the most adequate model to our problem, there is a need to explore which are the best parameters for each algorithm. To do so, we constructed graphs with the performances of different algorithm parameter combinations, with the final objective of finding trends and the sensitivity of the models.

The models were trained using 89159 entries, 10 predictors and 2 classes.

4.2.1 Knn

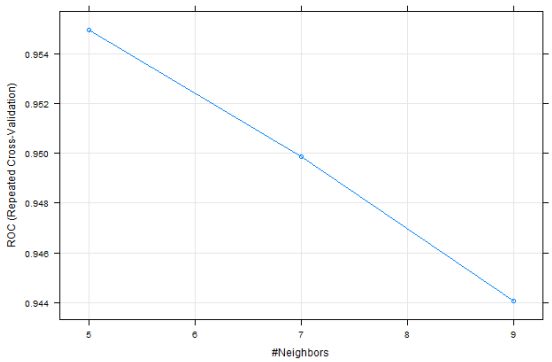
KNN is an extremely popular algorithm in classification that stores all available cases and classifies new cases based on a similarity measure to the previously saved ones. As we chose to use Euclidean distance as distance metric, the data was scaled between 0 and 1. The best choice of k depends of the data: in most cases, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct [Everitt et al., 2011]. We studied the effect of parameter *k* variation in the ROC and Sensitivity results, with the final objective of selecting the one who reached the highest scores on those metrics.

The results of the *train* function are displayed bellow. With the analysis of these values we can infer that both the ROC and Sensitivity values are higher the algorithm takes into consideration 7 neighbors to decide.

```
k-Nearest Neighbors

89159 samples
   10 predictor
    2 classes: 'churned', 'notchurned'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 80243, 80244, 80243, 80243, 80243, 80243, ...
Resampling results across tuning parameters:

  k  ROC        Sens       Spec
  5  0.9549515  0.8817442  0.8904896
  7  0.9498729  0.8693831  0.8769660
  9  0.9440586  0.8577111  0.8658240

ROC was used to select the optimal model using  the largest value.
The final value used for the model was k = 5.
```

On figure 4.1 the same results from the previous analysis are displayed. The graph shows the number of neighbors on the x axis and ROC values on the y axis. We can conclude that 5 is the number of neighbors that give the best ROC and Sensitivity results, and that value keeps decreasing with the increase on the number of neighbors.

Figure 4.1: Knn ROC value per number of neighbors



The training set class distribution was the following:

- churned: 38211
- not churned: 50948

After this step, new predictions were conducted using new data (test set) that was not used to train the model.

Using the knowledge acquired from the previous exploration, we chose 5 as the number of neighbors to create the final classification model.

A confusion matrix was then created based on the algorithm predictions with the new data. That confusion confusion matrix and associated statistics are listed bellow.

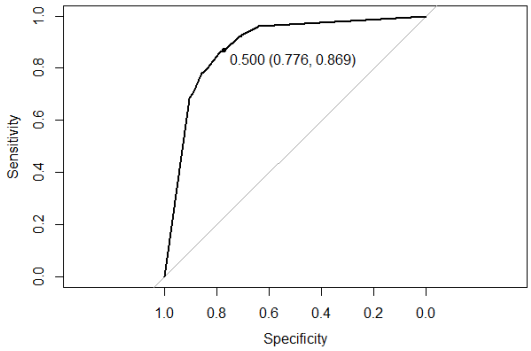
```
Confusion Matrix and Statistics

            Reference
Prediction   churned notchurned
  churned       4263       3295
  notchurned    1195      21265

               Accuracy : 0.8504
                 95% CI : (0.8463, 0.8544)
    No Information Rate : 0.8182
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5627
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.7811
            Specificity : 0.8658
         Pos Pred Value : 0.5640
         Neg Pred Value : 0.9468
             Prevalence : 0.1818
         Detection Rate : 0.1420
```

```
   Detection Prevalence : 0.2518
      Balanced Accuracy : 0.8234

       'Positive' Class : churned
```

Although we accomplished a high Specificity value, the Sensitivity value was considerably lower. And as already explained on section 4.1, Sensitivity has more relevance to our model.

We also acquired the ROC curve of this model, in order to obtain our second metric AUC. The ROC curve of this model is shown on figure 4.2. The AUC of this model was 0.8883.

Figure 4.2: Knn model ROC curve



4.2.2 Naive Bayes

Naive Bayes assumes that the value of a particular feature is independent from the value of any other feature, given the class variable.

Regarding this algorithm a study was conducted around the influence of the kernel choice in the models behavior.

When the "use kernel" value is set as TRUE, a kernel density estimate is used for density estimation; if it is set as FALSE a normal density is estimated.

The results of the this study are displayed bellow.

```
Naive Bayes

89159 samples
   10 predictor
    2 classes: 'churned', 'notchurned'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 80243, 80244, 80243, 80243, 80243, 80243, ...
Resampling results across tuning parameters:

  usekernel  ROC        Sens       Spec
  FALSE      0.6197166  0.6788183  0.4967745
   TRUE      0.7035849  0.5686231  0.7421816

Tuning parameter 'fL' was held constant at a value of 0
Tuning parameter 'adjust' was
 held constant at a value of 1
ROC was used to select the optimal model using  the largest value.
The final values used for the model were fL = 0, usekernel = TRUE and adjust = 1.
```

As we can deduce from the above results, the ROC value greatly increase when the option "usekernel" is activated, but the Sensitivity value decreases. We chose to keep the "usekernel" option as TRUE as this algorithms best model.

Using the values which gave the best results in the previous experiment, we conducted new predictions with that model on new data. In this way, we can truly evaluate the model behavior when tested with new values.

The confusion matrix values were:

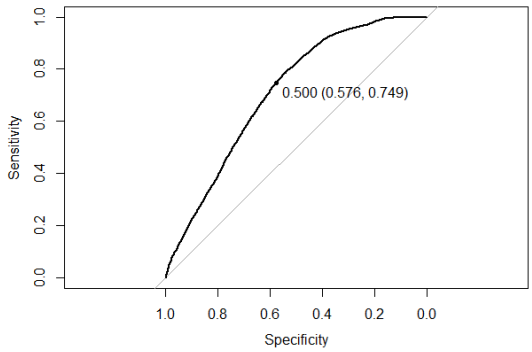
```
Confusion Matrix and Statistics

            Reference
Prediction   churned notchurned
  churned       3143       6173
  notchurned    2315      18387
```

```
               Accuracy : 0.7172
                 95% CI : (0.7121, 0.7223)
    No Information Rate : 0.8182
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.2545
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.5759
            Specificity : 0.7487
         Pos Pred Value : 0.3374
         Neg Pred Value : 0.8882
             Prevalence : 0.1818
         Detection Rate : 0.1047
   Detection Prevalence : 0.3103
      Balanced Accuracy : 0.6623

       'Positive' Class : churned
```

When tested on test data, this model continued to have the same behavior than in the train data: the Specificity value was extremely low, meaning that a lot of the clients who churned were not identified as such. This could have a negative impact on the company economy, and does not fulfill the requirements established in the beginning of this process.

The ROC curve on figure 4.3 shows that we obtain an AUC value of 0.7092 in the Naive Bayes model.

Figure 4.3: Naive Bayes ROC curve



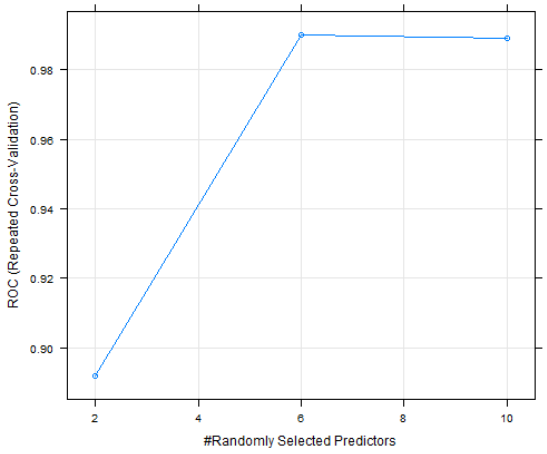
4.2.3 Random Forest

In the algorithm Random Forest, tunning parameter "mtry" represents the number of variables randomly sampled as candidates at each split. Multiple values of this parameter were tested to discover what value of the mtry parameter returns the most adequate model to our case study. 500 trees were constructed to train the model. The results of this research are shown bellow.

```
Random Forest

89159 samples
   10 predictor
    2 classes: 'churned', 'notchurned'

Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 80243, 80244, 80243, 80243, 80243, 80243, ...
Resampling results across tuning parameters:

  mtry  ROC        Sens       Spec
   2    0.8917940  0.5196755  0.9793253
   6    0.9900938  0.9050448  0.9789131
  10    0.9889823  0.9080370  0.9753343

ROC was used to select the optimal model using
  the largest value.
The final value used for the model was mtry = 6.
```

Although all the values for "mtry" return suitable models with good metric values, when tested with 6 predictors we get the higher values of ROC and Sensitivity. The same analysis can be drawn when analyzing figure 4.4. The maximum ROC value is hit when the number of predictors is 6; with the increase in the number of predictors, the results are slightly worse.

Figure 4.4: Random Forest number of predictors



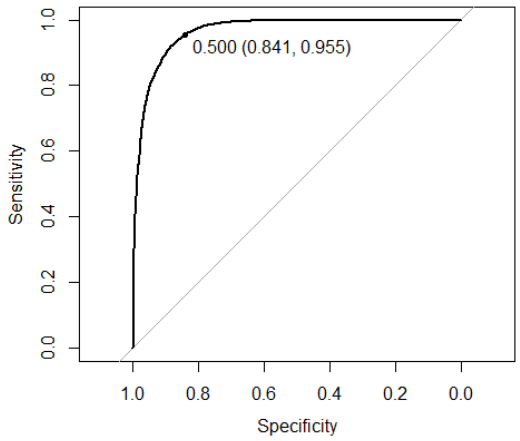
To confirm that this model can be considered a good one to our case study, there was a need to test it with new data. For that matter, a new prediction was conducted using the test set.

The generated confusion matrix from this prediction and its statistics are listed below:

```
Confusion Matrix and Statistics

            Reference
Prediction   churned notchurned
  churned       4594       1129
  notchurned     864      23431

               Accuracy : 0.9336
                 95% CI : (0.9307, 0.9364)
    No Information Rate : 0.8182
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.781
 Mcnemar's Test P-Value : 3.348e-09

            Sensitivity : 0.8417
            Specificity : 0.9540
         Pos Pred Value : 0.8027
         Neg Pred Value : 0.9644
             Prevalence : 0.1818
         Detection Rate : 0.1530
   Detection Prevalence : 0.1907
      Balanced Accuracy : 0.8979

       'Positive' Class : churned
```

When tested with the testing set, containing values not used to train the model, the model got a great score with a AUC of 0.9654 and a Sensitivity of 0.8417. The ROC curve is represented on figure 4.5. This model is proved to be a good fit in our study, with both of the chosen metrics to this research achieving good values.

Figure 4.5: Random Forest ROC curve



4.2.4 C4.5

C4.5 is an algorithm that has the objective of generating decision trees that can be used for classification. It has grown in popularity over the past few years [Wu et al., 2008] in the data mining community due to its good results when applied to data mining problems. J48 is an implementation of the C4.5 algorithm developed by the WEKA project team. R package "RWeka" contains this implementation, and was used to the development of this model. Our J48 model was trained with no parameter tunning. The results of the train are listed bellow.

```
C4.5-like Trees

89159 samples
   10 predictor
    2 classes: 'churned', 'notchurned'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 80243, 80244, 80243, 80243, 80243, 80243, ...
Resampling results:

  ROC        Sens       Spec
  0.9351585  0.8639919  0.9398733

Tuning parameter 'C' was held constant at a value of 0.25
```

Only with the cross-fold validation and the C value at 0.25, both the ROC and Sensitivity values are good.

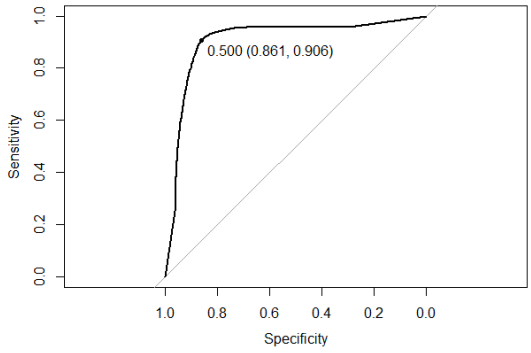
When tested with new data, the model continued to show a good performance. The model confusion matrix was the following:

```
Confusion Matrix and Statistics

            Reference
Prediction   churned notchurned
  churned       4700       2311
  notchurned     758      22249

               Accuracy : 0.8978
                 95% CI : (0.8943, 0.9012)
    No Information Rate : 0.8182
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.6906
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8611
```

```
            Specificity : 0.9059
         Pos Pred Value : 0.6704
         Neg Pred Value : 0.9671
             Prevalence : 0.1818
         Detection Rate : 0.1566
   Detection Prevalence : 0.2336
      Balanced Accuracy : 0.8835

       'Positive' Class : churned
```

The Sensitivity value is 0.86, which can be considered a good value in prediction. As we can see on figure 4.6, the model has a good performance with an AUC value of 0.9073.

Figure 4.6: J48 ROC curve



4.2.5 AdaBoost

Also called "Adaptive Boosting",it selects only those features known to improve the final predictive power of the model during the training stage. In this way, the execution time is potentially lowered, as the features that are not relevant to the problem are not computed.

On this algorithm model train, a study regarding which value combination of two tunning parameters returned the best model according to our evaluation. Those tunning parameters were number of trees to be generated (iter) and maximum tree depth (maxdepth). The results of the train model are shown bellow.

```
Boosted Classification Trees

89159 samples
   10 predictor
    2 classes: 'churned', 'notchurned'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 80243, 80244, 80243, 80243, 80243, 80243, ...
Resampling results across tuning parameters:

  maxdepth  iter  ROC        Sens       Spec
  1          50   0.6964185  0.4891696  0.8279684
  1         100   0.6972017  0.4577301  0.8756175
  1         150   0.7011062  0.4249996  0.9292482
  2          50   0.7218255  0.4145226  0.9364123
  2         100   0.7350789  0.4450194  0.9292741
  2         150   0.7463340  0.4518676  0.9283781
  3          50   0.7424751  0.4020829  0.9562692
  3         100   0.7677117  0.4133713  0.9722329
  3         150   0.7814356  0.4329902  0.9694395

Tuning parameter 'nu' was held constant at a value of 0.1
ROC was used to select the optimal model using  the largest value.
The final values used for the model were iter = 150, maxdepth = 3 and nu = 0.1.
```

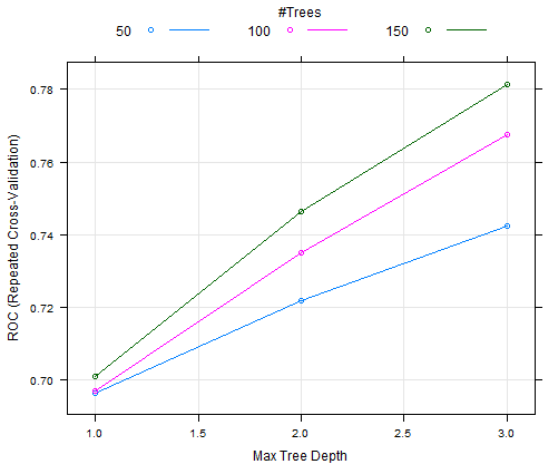
The combination of values that acquired the best ROC and Sensitivity results were maxdepth = 3 and iter = 150.

The same result can be drawn when analyzing figure 4.7. This figure shows the result of the AdaBoost train model. On the x axis is the maximum tree depth, and on the y axis the ROC values. Three different plot are represented, according to three different number of trees: 50, 100, 150.

The model previously chosen was then tested with new data, not used during the train phase. The generated confusion matrix was:

```
Confusion Matrix and Statistics
```

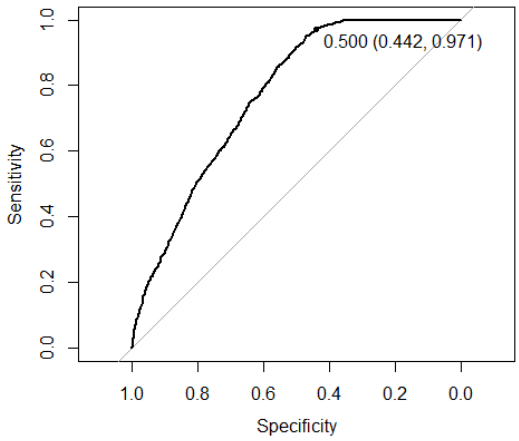
Figure 4.7: AdaBoost parameter study




```
            Reference
Prediction   churned notchurned
  churned       2412        715
  notchurned    3046      23845

               Accuracy : 0.8747
                 95% CI : (0.8709, 0.8784)
    No Information Rate : 0.8182
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.495
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.44192
            Specificity : 0.97089
         Pos Pred Value : 0.77135
         Neg Pred Value : 0.88673
             Prevalence : 0.18182
         Detection Rate : 0.08035
   Detection Prevalence : 0.10417
      Balanced Accuracy : 0.70640

       'Positive' Class : churned
```

The Sensitivity value from this model is considered to be low, with a value of 0.44. This has a negative effect both on the ROC value and on our expectations to our final desired model, because Sensitivity is a major classification value. The ROC curve of that model is represented on figure 4.8. As already mentioned, we can confirm that the Specificity values are always good, but the Sensitivity values are greatly worse. The final AUC value for this model is 0.7707.

Figure 4.8: AdaBoost ROC curve



4.2.6 ANN

In an ANN predictive model, the input layer contains all the inputs and variables used to predict the label variable, and the output layer contains the target of the prediction.

In this model, two parameters need to be fed to the algorithm: size and decay. The size value represents the number of units in the hidden layer, and the decay value is a parameter used in weight decay formula that penalizes solutions with high weights and bias.

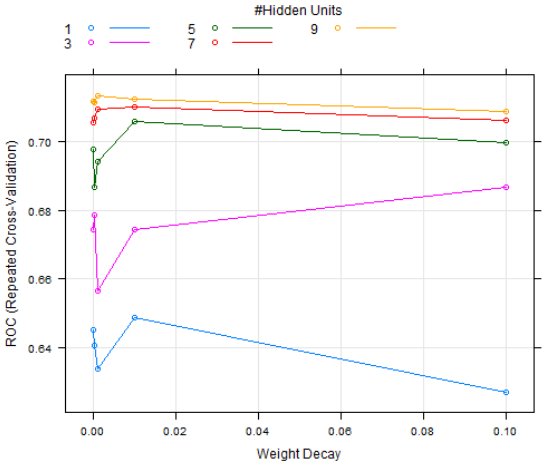
The train results are listed below.

```
Neural Network

89159 samples
   10 predictor
    2 classes: 'churned', 'notchurned'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 80243, 80244, 80243, 80243, 80243, 80243, ...
Resampling results across tuning parameters:

  size  decay  ROC        Sens       Spec
  1     0e+00  0.6451733  0.3811004  0.8745993
  1     1e-04  0.6407859  0.4583489  0.8067833
  1     1e-03  0.6337826  0.4545465  0.8034191
  1     1e-02  0.6489154  0.4576951  0.8012264
  1     1e-01  0.6271890  0.5508616  0.6974126
  3     0e+00  0.6743773  0.3714455  0.9255180
  3     1e-04  0.6785445  0.3701982  0.9337293
  3     1e-03  0.6565994  0.3833304  0.9166744
  3     1e-02  0.6742521  0.3972061  0.9026595
  3     1e-01  0.6866918  0.4200969  0.8730864
  5     0e+00  0.6977455  0.3696307  0.9461411
  5     1e-04  0.6866774  0.3830824  0.9340110
  5     1e-03  0.6939980  0.3737842  0.9326369
  5     1e-02  0.7056473  0.4169395  0.8950371
  5     1e-01  0.6996448  0.4258022  0.8712609
  7     0e+00  0.7055278  0.3889200  0.9309218
  7     1e-04  0.7066643  0.4113391  0.9066170
  7     1e-03  0.7093803  0.4073530  0.9089266
  7     1e-02  0.7099220  0.4115570  0.9023378
  7     1e-01  0.7062048  0.4466947  0.8512986
  9     0e+00  0.7115696  0.4035914  0.9164908
  9     1e-04  0.7113628  0.4120197  0.9093447
  9     1e-03  0.7132438  0.4272323  0.8943763
  9     1e-02  0.7120906  0.4343341  0.8806499
  9     1e-01  0.7085921  0.4576431  0.8393270

ROC was used to select the optimal model using  the largest value.
```

```
The final values used for the model were size = 9 and decay = 0.001.
```

With this result and through the analysis of figure 4.9, we can state that although the ROC values are acceptable, the Sensitivity values (considered to be important in our research) are considerably lower. In fact, not one of the parameter values combination reaches a Sensitivity over 0.50. In any case, we chose the value combination that reached the best ROC and Sensitivity results, which was size = 9 and decay = 0.001.

Figure 4.9: ANN parameter study



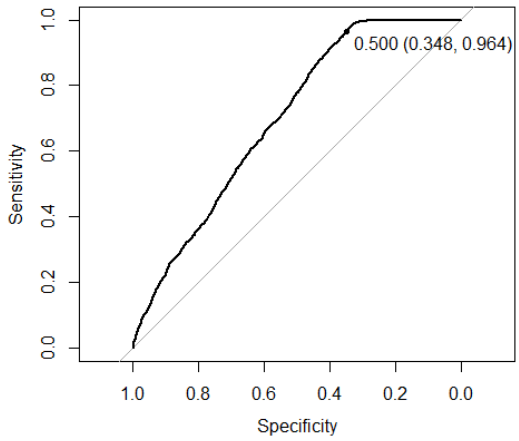
The next test was to test the model with data that was not used in the training stage. We conducted new predictions upon the test data, and created a confusion matrix to investigate the results. The final results of the new prediction are the following.

```
Confusion Matrix and Statistics

            Reference
Prediction   churned notchurned
  churned       1902        881
  notchurned    3556      23679

               Accuracy : 0.8522
                 95% CI : (0.8481, 0.8562)
    No Information Rate : 0.8182
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3862
 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.34848
            Specificity : 0.96413
         Pos Pred Value : 0.68344
```

```
         Neg Pred Value : 0.86943
             Prevalence : 0.18182
         Detection Rate : 0.06336
   Detection Prevalence : 0.09271
      Balanced Accuracy : 0.65630

       'Positive' Class : churned
```

As we had already determined in the training stage, the model does not have a good behavior in identifying the clients who churned. In fact, the Sensitivity result of the model with the test data was even lower than before (0.35). The ROC curve was also drawn to further investigate the model. Figure 4.10 is the visual representation of the curve to the ANN model. The analysis of this curve strengthens the already drawn conclusions from the previous results: an AUC of 0.697 is not sufficient to make this a good prediction model in customer churn.

Figure 4.10: ANN ROC curve



4.3 Model Comparison

The implemented models were already described and analyzed separately, but the main focus of this research was to identify which model can be considered the best to solve this problem. In order to make this statement, we need to truly compare all the trained models regarding our predefined metrics: AUC value and Sensitivity. This comparison is going to be done with the resource of the *caret* package also used to train the models. The functions that are going to be mentioned on this section were based on the research of [Hothorn et al., 2005] and [Eugster et al., 2012].

To make statistical statements about the algorithms performances and compare them, there was a need to collect their resampling results with the help of *resamples* function from package *caret*.

To guarantee that the models can be compared among themselves, there is a need to conduct hypothesis testing. A model with only the target variable and without any independent input variables can be called the null model (H0), as the model would be verified by the null hypothesis. By adding k input variables to create a fuller model (H1), we then can understand if the model is better with the input parameters or not.

We can define our H0 and H1 as such:

- **H1:** A customer's call pattern gives away its decision on churning.
- **H0:** A customer's call pattern its not connected with its decision on churning.

If we can not statistically prove H1, we must accept H0.

The p-value is the probability of observing data at least as favorable to the alternative hypothesis as our current data, considering that the null hypothesis(H0) is true. If the p-value is lower than 5%, it is very unlikely (with < 5% probability) that the null hypothesis is actually true when we rejected it. So, when the p-value crosses the 5% threshold, we can reject the null hypothesis (H0) in favor or the alternative hypothesis (H1). On table 4.2 are the p-values of the models retrieved when the confusion matrix for each models was analyzed. All the models conquered values under the predefined threshold of 0.05, so we can discard H0 and accept H1.

Table 4.2: Models P-values

Model	**p-value**
KNN	<2.2e-16
NB	<2.2e-16
J48	<2.2e-16
RF	<2.2e-16
ADA	<2.2e-16
ANN	<2.2e-16

This section will analyze and compare the algorithms regarding three values: AUC (named ROC in the following sections), Sensitivity, and Specificity. We will consider Sensitivity as a more important value than Specificity as already explained.

Table 4.3: AUC value comparison

	Min.	1st Qu.	Median	Mean	3rd	Qu.	NA's
KNN	0.9509	0.9534	0.9544	0.9550	0.9566	0.9599	0
NB	0.6866	0.6991	0.7045	0.7036	0.7078	0.7170	0
J48	0.9276	0.9327	0.9353	0.9352	0.9369	0.9446	0
RF	0.9882	0.9898	0.9902	0.9901	0.9907	0.9915	0
ADA	0.7677	0.7771	0.7812	0.7814	0.7860	0.7970	0
ANN	0.6895	0.7082	0.7128	0.7132	0.7190	0.7318	0

The AUC is a common evaluation metric for binary classification problems. It represents the area value created by the ROC curve. If a classifier is very good, the true positive rate will increase quickly and the area under the curve will be close to 1. If the classifier is similar to random guessing, the true positive rate will increase linearly with the false positive rate and the area under the curve will be around 0.5. The values regarding AUC for our trained model are on table 4.3. Among the studied models, three of them achieved very good AUC values: KNN, J48 and and Random Forest.

Figure 4.11: ROC values box plot

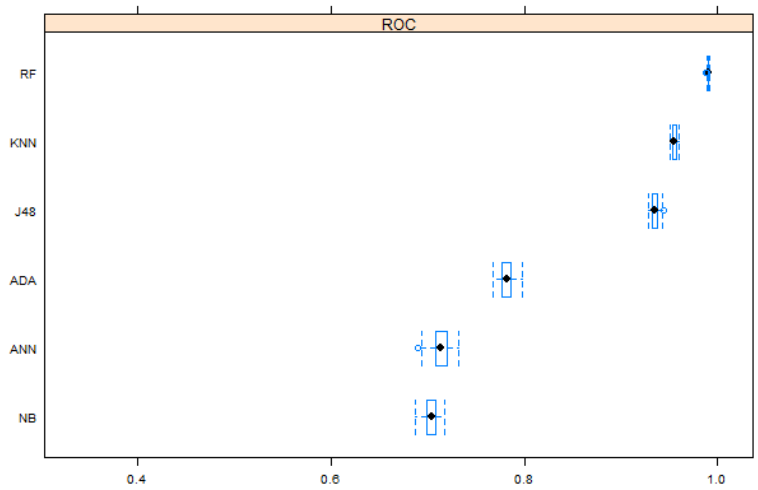


Figure 4.11 is a visual representation of the model's ROC values trough a box plot. This kind of visualization enables the extraction of new knowledge. The algorithms who reached the lowest ROC scores are also the ones with the highest distance between minimum and maximum values.

Regarding Sensitivity values displayed on table 4.4, the results were worse than in the previous metric. The same models that had the higher AUC values are still on the top Regarding Sensitivity values. The AdaBoost and ANN models have very poor Sensitivity values, and can be considered not a good fit to our problem.

When analyzing the sensitivity values of the generated models in a box plot in figure 4.12, the same conclusion from before can be drawn. The worse models also have the highest disparity in

Table 4.4: Sensitivity value comparison

	Min.	1st Qu.	Median	Mean	3rd	Qu.	NA's
KNN	0.8710	0.8793	0.8818	0.8817	0.8854	0.8930	0
NB	0.5475	0.5651	0.5691	0.5686	0.5751	0.5861	0
J48	0.8448	0.8610	0.8650	0.8640	0.8686	0.8752	0
RF	0.8974	0.9017	0.9054	0.9050	0.9084	0.9110	0
ADA	0.4038	0.4142	0.4346	0.4330	0.4464	0.4715	0
ANN	0.3481	0.3860	0.4372	0.4272	0.4575	0.4847	0

their values. In this case, this is specially visible in ANN algorithm.

With the previous analysis we acquired an overview of all the models. But to make it simpler to compare them, and since they were fit on the same versions of the training data, we are able to make inferences on the differences between models. In this way we reduce the within-resample correlation that may exist. The results of the study on models differences is shown bellow. The upper diagonal values of the tables represent estimates of differences between models, and the lower-diagonal are values representing a one-sided test to see if the accuracy is better than the "no information rate," which is taken to be the largest class percentage in the data. With these values, we can investigate whether the accuracy of one model is better than the proportion of the data with the majority class.

```
p-value adjustment: bonferroni
Upper diagonal: estimates of the difference
Lower diagonal: p-value for H0: difference = 0

ROC
    KNN       NB        J48       RF        ADA       ANN
KNN            0.251367  0.019793 -0.035142  0.173516  0.241708
NB  < 2.2e-16            -0.231574 -0.286509 -0.077851 -0.009659
J48 < 2.2e-16 < 2.2e-16           -0.054935  0.153723  0.221915
RF  < 2.2e-16 < 2.2e-16 < 2.2e-16            0.208658  0.276850
ADA < 2.2e-16 < 2.2e-16 < 2.2e-16 < 2.2e-16            0.068192
ANN < 2.2e-16 5.761e-05 < 2.2e-16 < 2.2e-16 < 2.2e-16
```

As the p-values for all the pairs are under 0.05, we can compare the algorithms.

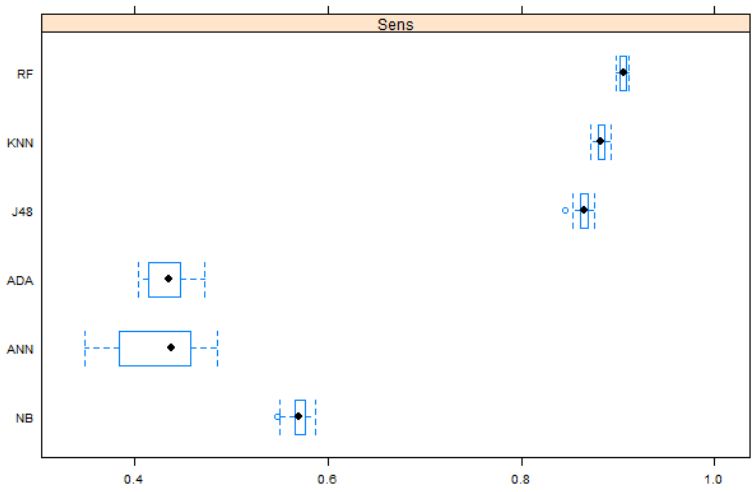
Regarding AUC values, the greatest difference between two models is with Naive Bayes and Random Forest, the worst and best models respectively.

The ones that distinguish themselves for the bad results are mostly ANN and Naive Bayes.

To visualize this value differences, we created a box plot represented in figure 4.13. The KNN, J48 and random forest models have almost none variation between themselves. The highest gap is between the Random forest and ANN models.

```
Sens
```

Figure 4.12: Sensitivity values box plot




```
    KNN        NB         J48        RF         ADA        ANN
KNN             0.313121   0.017752 -0.023301   0.448754   0.454512
NB  < 2.2e-16              -0.295369 -0.336422   0.135633   0.141391
J48 3.753e-12 < 2.2e-16               -0.041053   0.431002   0.436760
RF  < 2.2e-16 < 2.2e-16 < 2.2e-16                0.472055   0.477812
ADA < 2.2e-16 < 2.2e-16 < 2.2e-16 < 2.2e-16                 0.005758
ANN < 2.2e-16 < 2.2e-16 < 2.2e-16 < 2.2e-16 < 2.2e-16
```

These second results confirm the drawn conclusions from before. It is also possible to delineate two distinct groups with high similarity within the group: KNN, J48 and random forest ; and naive bayes, adaBoost and ANN.

Figure 4.14 represents this values but for all the trained models. The highest difference values is clearly between random forest and ANN, possibly due to their results difference and ANN model variation.

Figure 4.13: ROC values differences box plot

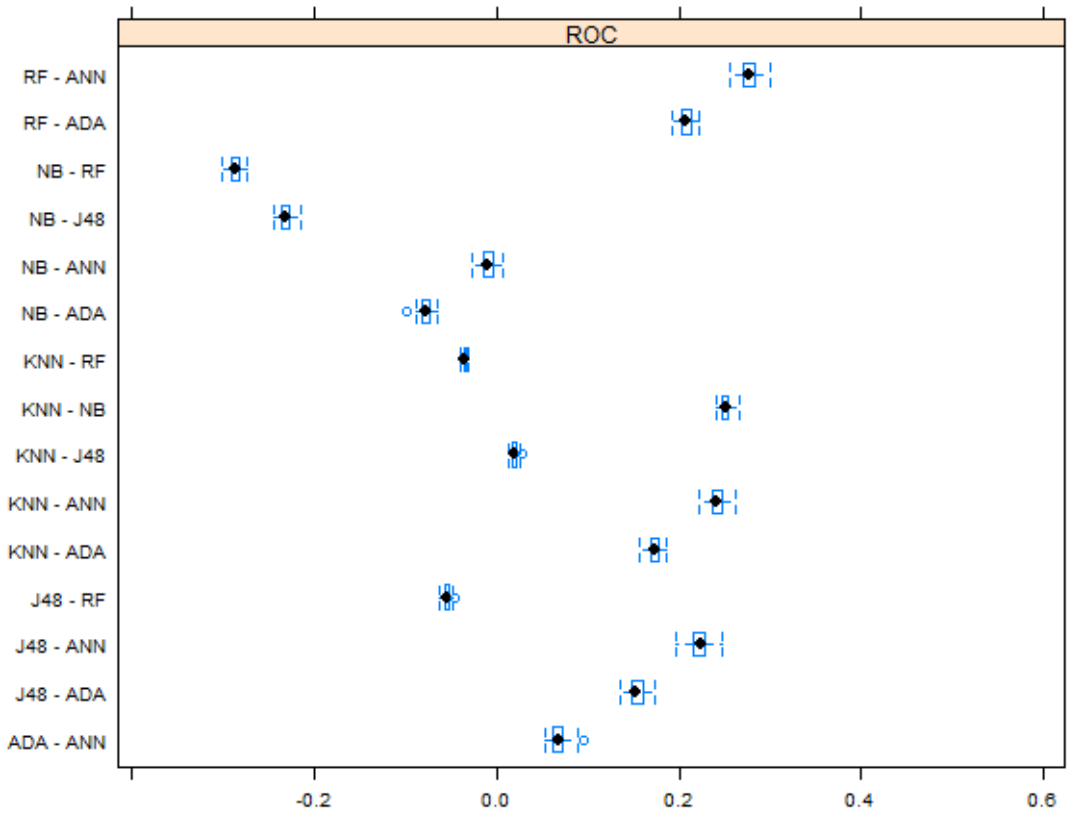
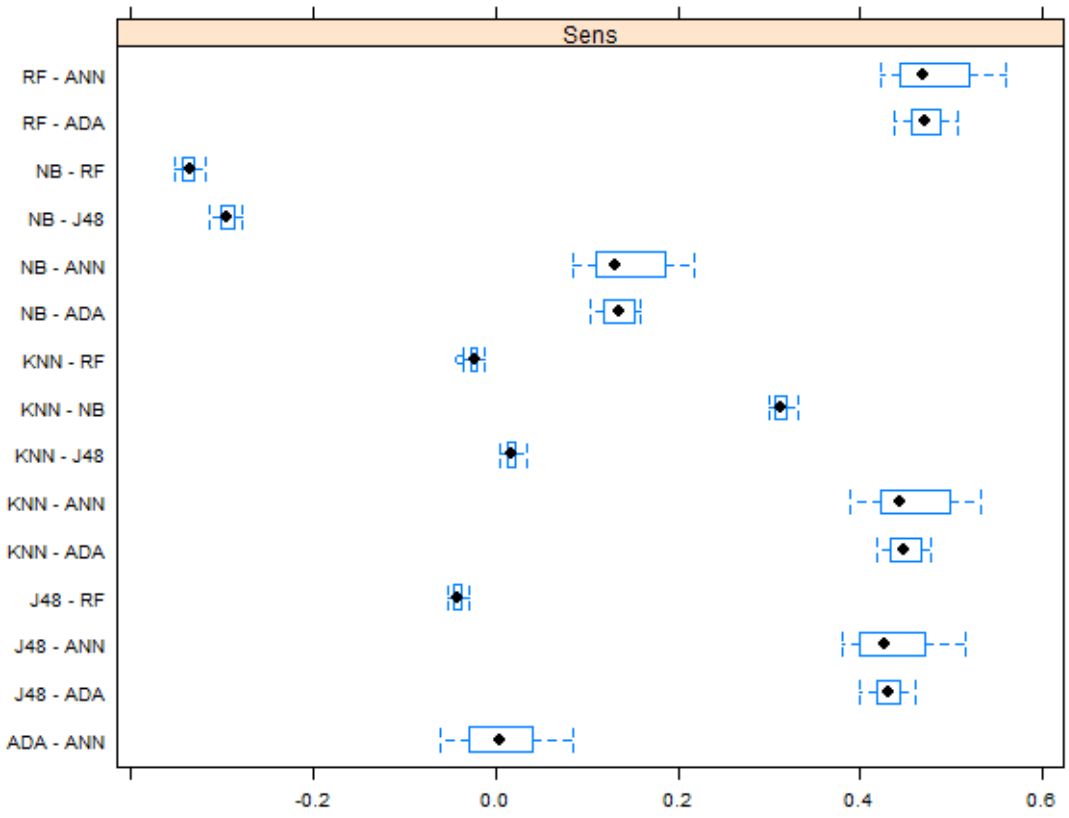


Figure 4.14: Sensitivity values differences box plot



4.3.1 Variable importance

All the models were constructed based on the available data variables. However, we did not had the information regarding their relative importance to our models.

We conducted a study to acquire this values for the Random Forests method, to assess which predictors had the largest impact on the model which got the best results.

The results of this study are presented on table 4.5. All measures of importance were scaled to have a maximum value of 100.

Variable "duration" has clearly the most importance to our model, setting the top value of 100. The next variables significant to our model are the call "direction" and a number representing the other person on the call.

Table 4.5: ROC curve variable importance

Variable	**Importance**
duration	100.000
direction	61.275
other_msisdin	55.965
start_cell_id	29.924
dropped_call	24.278
destination_type	24.205
voicemail	22.547
end_cell_id	12.009
call_type	4.723
other_contract_id	0.000

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The importance of this type of research to the telecom market is continuously growing. Data collecting is becoming an everyday task to all companies, and the value of that data can come from multiple sources. Churn prediction is becoming one of those sources that create revenue to the company. Being able to prevent when clients are going to cease their contract with the company opens the possibility of renegotiating that contract in order to retain the costumer.

This dissertation aimed to create suitable models that predict customer churn. This models needed to register high values in the defined metrics: AUC and Sensitivity. To validate the models, we chose to implement 10-folds cross validation with 3 repeats.

The dataset was too extensive to be used in the available time to complete this dissertation, so there was a need to reduce its dimension. We took into consideration it was more important to have the full history of a few clients instead of one month history of all the clients.

The data itself was not suitable to predict upon. It had to go through multiple transformations to make it fit to train and test the models.

As the database was provided by a company with their real values, it did not contained the information regarding the target variable: if a costumer had churned or not. According to the standard in this circumstances, we set that variable according to a simple rule: if an account did not make or receive any calls during a 30 day time period, we considered that client has cease the contract. However, when analyzing this new variable we came across another problem: it was imbalanced. Only around 18% of the entries represented churning clients. If our models were trained upon this data, it could result in overfitted models that disregard the minority class. Our approach to solve this problem consisted on applying an hybrid sampling method SMOTE. Our final dataset was much more balanced than before with the churned class represented by 0.43% of all entries, leading to balanced models.

Six models were trained and tested. The algorithms themselves were chosen due to their diversity and applicability in this kind of prediction. Those algorithms were KNN, naive bayes, random forest, C4.5, AdaBoost and ANN.

Random forest model achieved the best results in both ROC and Sensitivity in all measurements. With a highest ROC value of 0.9915 and Sensitivity value 0.9110, it distinguishes itself from the other models.

KNN and J48 models come in second and third place respectively. They acquired similar results to the random forest model, being their AUC and Sensitivity values slightly lower.

The remaining algorithms, (naive bayes, ada and ann) failed to positively predict upon our data. Their ROC values were acceptable , but their Sensitivity values are considerably low. This means that this algorithms did not conceived good models according to our metrics and needs.

When compared with the reviewed literature our chosen model achieved better results. In the Buckinx and Poel [Buckinx and den Poel, 2005] study, the model that reached the highest values was also the random forest algorithm, but with an AUC value in the testing data of 0.8319.

With this dissertation we were able to prove that the classification models applied to the available dataset were a good fit and produced good models to predict customer churn.

As the dataset used in this research represent real calls from a telecom company, this model can effectively be used to the company's welfare. The administration, using this kind of model, can predict and target the clients that are close of ceasing their contract.

5.2 Future Work

Although we consider the conducted study a success, we acknowledge that there is still room for improvements.

- Since our dataset timeframe is 7 months (difference between first and last entry), some of the older data can influence in a negative way the models. It would be interesting to investigate which data timeframe produce the best result and how the efficiency of the models is influenced by this timeframe. We could base our approach on the studies conducted by some authors [Ballings and Van Den Poel, 2012] in similar conditions.

- Due to computational limitations we were only able to use a small percentage of the available dataset. The remaining dataset could be used in two ways to improve our research: use all the data to predict and train the models; or select all the churning entries and the same amount of non churners to prevent class imbalance. On this second approach, we would not need to apply sampling methods to the data.

- The dataset available to this research has a great size regarding number of entries. However, the number of available variables to predict upon are quite limited and we had no demographic information, considered to be important to this kind of prediction. It would be of great interest to add these variables to our data and see if they improve the models. This

will also open up the possibility of conducting input selection based on the model's variable importance [Verbeke et al., 2012], removing irrelevant variables from the dataset that can have a negative effect on some of the algorithms (e.g., knn, ann).

- The objective of this dissertation was to compare the behavior of multiple algorithms in the conditions. Due to computational and time restrictions, we chose 7 of the most popular and suitable algorithms to construct the models. It would be very interesting to expand that number, increasing the number of applied classification algorithms.

- All the algorithms suffered parameter tunning to identify which parameter values returned the best models. But there are many other approaches to tune the models that were not tested and could improve the final models.

- After exploring many evaluation metrics, AUC and Sensitivity values were believed to be the most relevant ones to our research. It would be interesting to evaluate the model's behavior with different metrics.

References

[Ascarza et al., 2016] Ascarza, E., Iyengar, R., and Schleicher, M. (2016). The perils of proactive churn prevention using plan recommendations: Evidence from a field experiment. *Journal of Marketing Research*, 53(1):46–60.

[Balle et al., 2013] Balle, B., Casas, B., Catarineu, A., Gavaldà, R., and Manzano-Macho, D. (2013). The architecture of a churn prediction system based on stream mining. *Frontiers in Artificial Intelligence and Applications*, 256:157–166.

[Ballings and Van Den Poel, 2012] Ballings, M. and Van Den Poel, D. (2012). Customer event history for churn prediction: How long is long enough? *Expert Systems with Applications*, 39(18):13517–13522.

[Bott, 2014] Bott, R. (2014). Predicting Customer Churn in Telecom Industry using Multilayer Preceptron Neural Networks: Modeling and Analysis. *Igarss 2014*, 11(1):1–5.

[Buckinx and den Poel, 2005] Buckinx, W. and den Poel, D. V. (2005). Customer base analysis: partial defection of behaviourally loyal clients in a non-contractual {FMCG} retail setting. *European Journal of Operational Research*, 164(1):252 – 268.

[Chawla et al., 2002] 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.

[Coussement and Poel, 2008] Coussement, K. and Poel, D. V. d. (2008). Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert Systems with Applications*, 34(1):313 – 327.

[CRAN,] CRAN. caret: Classification and Regression Training. https://cran.r-project.org/web/packages/caret/index.html. [Online; Accessed May 15, 2016].

[C.Saranya, 2013] C.Saranya, G. (2013). A study on normalization techniques for privacy preserving data mining. *International Journal of Engineering and Technology*.

[Eugster et al., 2012] Eugster, M. J. A., Hothorn, T., and Leisch, F. (2012). Domain-Based Benchmark Experiments : Exploratory and Inferential Analysis. *AUSTRIAN JOURNAL OF STATISTICS*, 41(1):5–26.

[Everitt et al., 2011] Everitt, B., Landau, S., Leese, M., and Stahl, D. (2011). *Cluster Analysis*. Wiley Series in Probability and Statistics. Wiley.

[Fern and Cernadas, 2014] Fern, M. and Cernadas, E. (2014). Do we Need Hundreds of Classifiers to Solve Real World Classification Problems ? *Journal of Machine Learning Research*, 15:3133–3181.

[FEUP,] FEUP. GridFEUP. https://www.grid.fe.up.pt/. [Online; Accessed May 20, 2016].

[Foundation,] Foundation, T. R. The R Project for Statistical Computing. https://www.r-project.org/. [Online; Accessed March 19, 2016].

[Gerpott et al., 2001] Gerpott, T. J., Rams, W., and Schindler, A. (2001). Customer retention, loyalty, and satisfaction in the German mobile cellular telecommunications market. *Telecommunications Policy*, 25(4):249–269.

[Han et al., 2012] Han, S. H., Lu, S. X., and Leung, S. C. H. (2012). Segmentation of telecom customers based on customer value by decision tree model. *Expert Systems with Applications*, 39(4):3964–3973.

[Hothorn et al., 2005] Hothorn, T., Leisch, F., and Hornik, K. (2005). The Design and Analysis of Benchmark Experiments. 14(3).

[Kim and Yoon, 2004] Kim, H. S. and Yoon, C. H. (2004). Determinants of subscriber churn and customer loyalty in the Korean mobile telephony market. *Telecommunications Policy*, 28(9-10):751–765.

[Kim et al., 2004] Kim, M. K., Park, M. C., and Jeong, D. H. (2004). The effects of customer satisfaction and switching barrier on customer loyalty in Korean mobile telecommunication services. *Telecommunications Policy*, 28(2):145–159.

[Kuhn, 2008] Kuhn, M. (2008). Building predictive models in r using the caret package. *Journal of Statistical Software*.

[Ngai et al., 2009] Ngai, E., Xiu, L., and Chau, D. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, 36(2 PART 2):2592–2602.

[Oentaryo et al., 2012] Oentaryo, R. J., Lim, E.-p., Lo, D., Zhu, F., and Prasetyo, P. K. (2012). Collective Churn Prediction in Social Network.

[Oghojafor et al., 2012] Oghojafor, B. E. A., Mesike, G. C., Omoera, C. I., and Bakare, R. D. (2012). Modelling telecom customer attrition using logistic regression. *African Journal of Marketing Management*, 4(3):110–117.

[Olaleke et al., 2014] Olaleke, O., Borishade, T., Adeniyi, S., and Omolade, O. (2014). Empirical analysis of marketing mix strategy and student loyalty in education marketing. *Mediterranean Journal of Social Sciences*, 5(23):616–625.

[R Core Team, 2013] R Core Team (2013). R: A language and environment for statistical computing. [Online; Accessed March 19, 2016].

[Radosavljevik et al., 2010] Radosavljevik, D., Putten, P. V. D., and Larsen, K. K. (2010). The Impact of Experimental Setup in Prepaid Churn Prediction for Mobile Telecommunications : What to Predict , for Whom and Does the Customer Experience Matter ? 3(2):80–99.

REFERENCES

[Rahm and Do, 2000] Rahm, E. and Do, H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin*, 23:3–13.

[Reinartz and Kumar, 2002] Reinartz, W. and Kumar, V. (2002). The Mismanagement of Customer Loyalty. *Harvard Business Review*.

[Retana et al., 2016] Retana, G., Forman, C., and Wu, D. (2016). Proactive customer education, customer retention, and demand for technology support: Evidence from a field experiment. *Manufacturing and Service Operations Management*, 18(1):34–50.

[Shao and Deng, 2016] Shao, H. and Deng, X. (2016). Short-term wind power forecasting using model structure selection and data fusion techniques. *International Journal of Electrical Power and Energy Systems*, 83:79 – 86.

[Software,] Software, T. Tableau. http://www.tableau.com/. [Online; Accessed April 2, 2016].

[SQLite,] SQLite. SQLite. https://www.sqlite.org/. [Online; Accessed March 18, 2016].

[Tsai and Lu, 2009] Tsai, C. F. and Lu, Y. H. (2009). Customer churn prediction by hybrid neural networks. *Expert Systems with Applications*, 36(10):12547–12553.

[Verbeke et al., 2012] Verbeke, W., Dejaeger, K., Martens, D., Hur, J., and Baesens, B. (2012). New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. *European Journal of Operational Research*, 218(1):211–229.

[WeDoTechnologies,] WeDoTechnologies. Assuring your business for the future. http://www.wedotechnologies.com/pt/. [Online; Accessed March 1, 2016].

[Wei and Chiu, 2002] Wei, C. P. and Chiu, I. T. (2002). Turning telecommunications call details to churn prediction: A data mining approach. *Expert Systems with Applications*, 23(2):103–112.

[Weston and Calaway, 2015] Weston, S. and Calaway, R. (2015). Getting Started with doParallel and foreach. pages 1–6.

[Wu et al., 2008] Wu, X., Kumar, V., Ross, Q. J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z. H., Steinbach, M., Hand, D. J., and Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*.

[Yen and Wang, 2006] Yen, C. and Wang, H.-y. (2006). Applying data mining to telecom churn. 31:515–524.

[Zaki and Meira, 2013] Zaki, M. J. and Meira, M. J. (2013). *Data Mining and Analysis: Fundamental Concepts and Algorithms*.

[Zaki and Wong, 2003] Zaki, M. J. and Wong, L. (2003). Data mining techniques. *WSPC*.

REFERENCES

Appendix A

Results

Figure A.1: Specificity values box plot

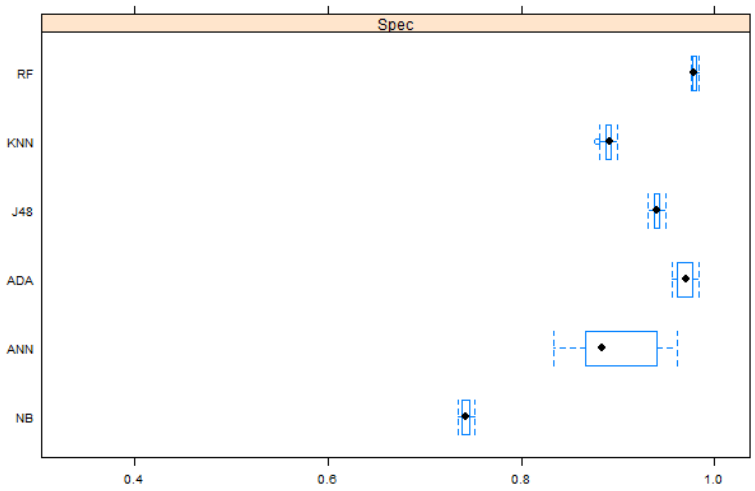


Figure A.2: ROC values dot plot

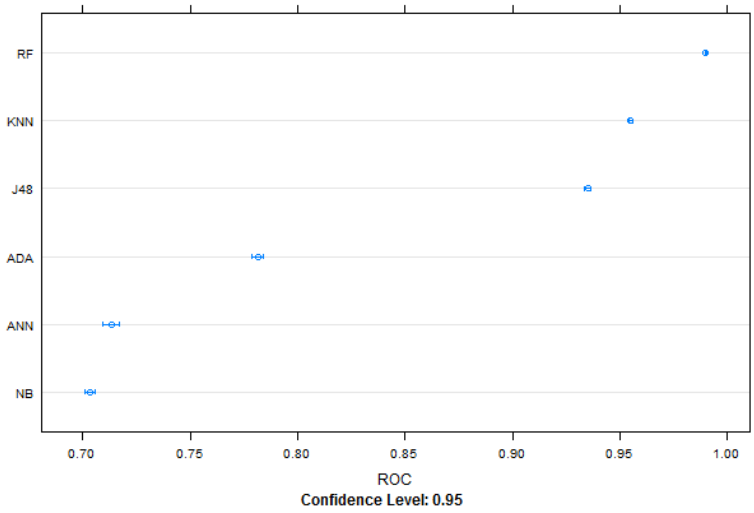


Figure A.3: Specificity comparison values box plot

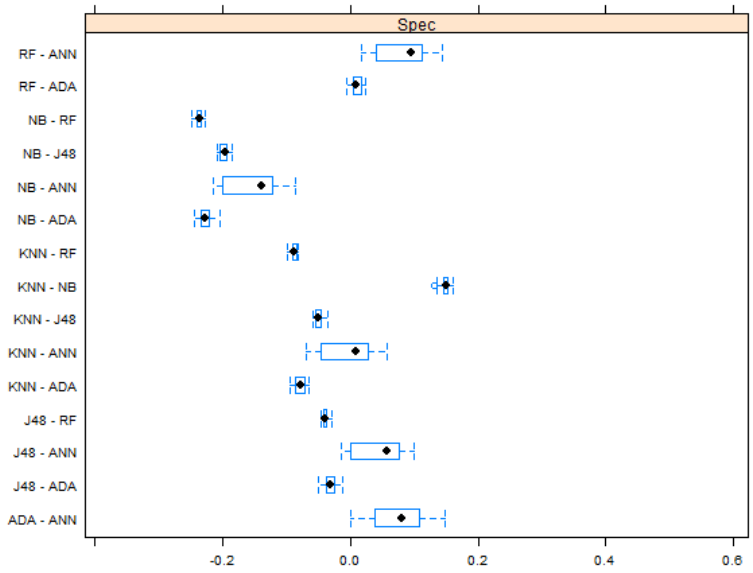


Figure A.4: Model comparison dot plot

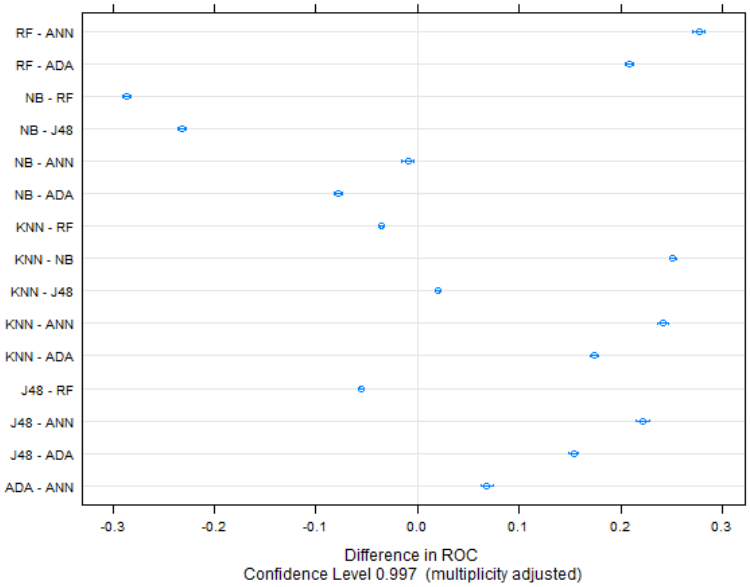
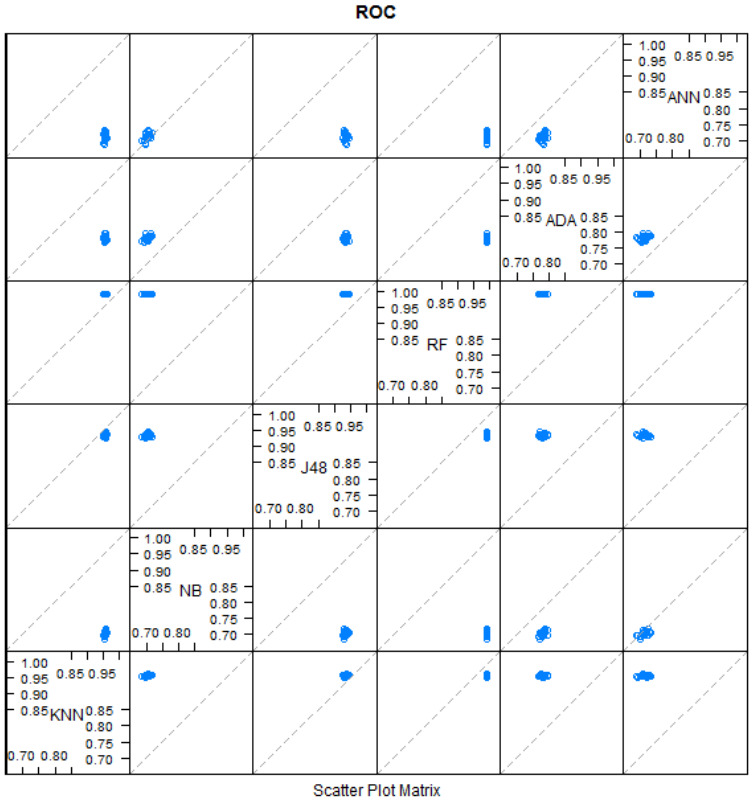


Figure A.5: ROC values scatter plot matrix



Appendix B

Paper

Churn Prediction in the Telecom Business

Georgina Esteves and João Moreira

Faculdade de Engenharia, Universidade do Porto
Rua Dr. Roberto Frias, s/n
Porto, Portugal 4200-465,
ei10010@fe.up.pt jmoreira@fe.up.pt,

Abstract. Telecommunication companies are acknowledging the existing connection between customer satisfaction and company revenues. Customer churn in telecom refers to a customer that ceases his relationship with a company. Churn prediction in telecom has recently gained substantial interest of stakeholders, who noticed that retaining a customer is substantially cheaper that gaining a new one. This research compares six approaches that identify the clients who are closer to abandon their telecom provider. Those algorithms are: KNN, Naive Bayes, J48, Random Forest, AdaBoost and ANN. The use of real data provided by WeDo technologies extended the refinement time necessary, but ensured that the developed algorithm and model can be applied to real world situations. The models are evaluated according to three criteria: are under curve, sensitivity and specificity, with special weight to the first two values. The Random Forest algorithm proved to be the most adequate in all the test cases.

Keywords: Churn Prediction; Telecom; Machine Learning; Churn Analysis; Customer attritions analysis; Customer attritions

1 Introduction

Since the 1990s the telecommunications sector became one of the key areas to the development of industrialized nations. The main boosters were technical progress, the increasing number of operators and the arrival of competition. This importance has been accompanied by an increase in published studies sector and more specifically on marketing strategies [1].

In order to acquire new costumers, a company must invest significant resources to provide a product or service that stands out from the competitors, however, continuously evolving the product itself is not enough. These companies are realizing that a customer-oriented business strategy is crucial for sustaining their profit and preserving their competitive edge [4]. As acquiring a new costumer can add up to several times the cost of efforts that might enable the firms to retain a customer, a best core marketing strategy has been followed by most in the telecom market: retain existing customers, avoiding customer churn [2, 3].

According to the author [5], 'costumer churn' in telecom business refers to the costumer movement from one provider to another. 'Customer management' is the process conducted by a telecom company to retain profitable costumers.

The objective of this research is to develop a method for churn prediction in the telecom business. The intended result is an algorithm that identifies the clients that are closer to abandon their current operator. Although many approaches were conducted in the past few years, there are still many opportunities to improve the current work in this area. The data that will be used during this work is fundamental to assure the quality of the final solution. The dataset is an extend collection of calls conducted by real customers. This will extend the refinement time necessary, but will ensure that the developed algorithm and model can be applied to real world situations. It opens a new set of possibilities, and makes it possible to obtain interesting and novel results.

2 Dataset

The models that predict customer churn are based on knowledge regarding the company's clients and their calls. That information is stored in a database table and is called dataset.

All the models were trained and tested with this data. But before that can happen, this collection had to go through multiple transformations and preprocessing techniques to make it suitable to predict upon.

2.1 Data collection

For the purpose of this research, real data from WeDo technologies client calls was provided by the company. The data is stored in a SQL file, and has a size of more than 131 gigabytes. The file contains one table with over 1.2 billion entries from 5 million different clients.

2.2 Data selection

Due to computational and time limitations, the dataset available to this research was too large and needed to be sampled. In order to prevent the lost of valuable information and keep the final results accurate, a simple selection of the first N table entries was not viable. The right approach to this problem is to retrieve all the call records from a group of clients. The final dataset contained over 100 thousand calls between 30 June 2012 and 31 January 2013.

2.3 Data analysis

The dataset available to this research contains information regarding call information of telecom company clients.

Table 1 contains an overview of the data. It identifies the variables, their values and a simple description of their meaning.

Figure 1 is a visual representation of the duration in minutes of client calls. The calls were group in 20 minutes intervals. The graph was also trimmed at 1000 minutes to make the graph more easily understandable. We can conclude

Table 1. Variables in the data

Variable	Value	Description
DATE	YearMonthDay	Date of the call
TIME	HoursMinutesSeconds	Time of the call
DURATION	Seconds	Call duration
MSISDN	Numeric	Anonymized number. If Incoming, it is the number getting the call. If Outgoing, is the number calling.
OTHER_MSISDN	Numeric	The"other" number in the call (regarding the previous variable).
CONTRACT_ID	Numeric	Client code
OTHER_CONTRACT_ID	Numeric	Not present in all data (does not belong to the operator).
START_CELL_ID	Numeric	Should represent the calling device (Outgoing).
END_CELL_ID	Numeric	Should represent the device getting the call (Ingoing)
DIRECTION	I, O	Represent a incoming call,("I") or outgoing,("O").
CALL_TYPE	FI, MO, ON, OT, SV	FI - The other device in the call bellongs to a wireline MO - The other device in the call bellongs to amobile network ON - On-Net, both incoming and outgoing systems bellong to this operator OT - Others SV - Services:VoiceMail calls, etc
DESTINATION_TYPE	I, L	Local (L) or international (I)
DROPPED_CALL	Y, N	Dropped call (Yes or No)
VOICMAIL	Y, N	Call went to voicemail (Yes or No)

that the majority of the calls on our dataset have a duration between 1 and 60 minutes. This number continues decreasing with the growth of the call duration, and stabilizes near 440 minutes.

In our dataset, variable "direction" expresses if a call record regards an incoming call ("I") or an outgoing call ("O"). The majority of the calls represent an outgoing call, and only around 35 thousand are incoming calls.

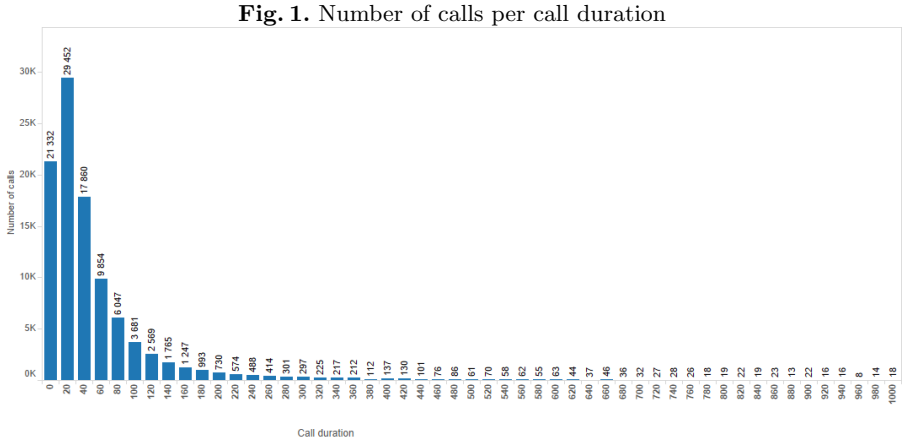
Around 90% of the calls belong to the Mobile and On-Net groups.

The majority of the calls are between numbers from the same country, named local (L) calls. Only 3% of the calls in our dataset are international (I). A similar result can be found when analyzing dropped calls data. A call is said to be dropped when due to technical reasons, is cut off before the speaking parties had finished their conversation and before one of them had hung up. Only around 2% of our data has this specific characteristic.

But the main problem regarding our data was the lack of knowledge of what costumers churned and the ones that did not. According to some authors [6, 7], we can state that a costumer has churned a telecom company when he does not do or receive any communication during 30 days or more.

The respective calculations were conducted, and final results are shown in figure 2. More than 60 % of clients have done communications in less then 5 days, which tells they are currently active. We also have an estimate of approximately 26% customers that churned the company, with a maximum day difference of 210 days without communications.

Fig. 1. Number of calls per call duration



The final distribution of the clients concerning churn had a 16% churn to 84% not churn ratio. This is our target variable, which means that the objective of the developed models is to predict the outcome of this variable through the study of the other variables in our data. The problem regarding imbalanced class will be addressed in the next chapter.

3 Experimental Setup

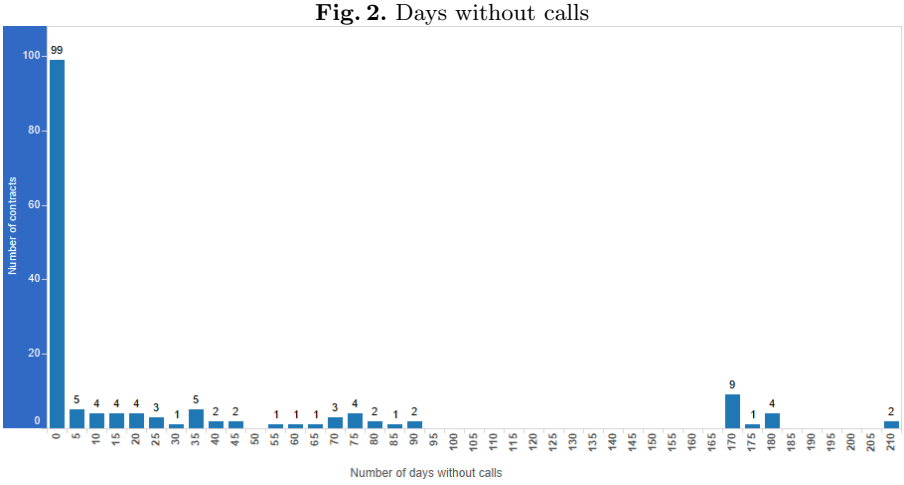
To guarantee the results integrity when comparing multiple algorithms, we need to assure that all the algorithms are tested in the same conditions and with the same data. As the original data was ordered, we applied a randomize function to the dataset. We also removed variables "contract_id" and "msisdn" from the data that was going to train the multiple models, because they have a direct connection with the outcome of target variable (identify the client itself).

The solution found consists in dividing the original data into two subsets: train set and test set with a 70/30 ratio.

Has we have already mentioned on the previous section, our target class is imbalanced, and this value disparity could have a significant negative impact on the final models regarding model fitting. The chosen approach was a hybrid sampling method [8] named Synthetic Minority Over-sampling Technique (SMOTE) that was applied to the training set. This method approaches the data in two ways: generates new examples of the minority class using the nearest neighbors of that cases, and under-samples the majority class.

The evaluation metrics chosen are Area Under Curve (AUC), sensitivity and specificity. In customer churn analysis it might be more expensive to incorrectly infer that customer is not churning then to give a general reduction in prices for services to clients that are not planning to leave the company. Since the priority in our case study is given to identifying churn clients rather than not churn ones, Sensitivity is more relevant than Specificity in our results.

Fig. 2. Days without calls



The algorithms applied were chosen due to their diversity of representation and learning style, and their common application on this kind of problems. We also took into consideration studies regarding the popularity and efficiency [9].

Six different machine learning models were trained and compared among themselves. Those algorithms were:

- Knn
- Naive Bayes
- Random Forest
- C4.5
- AdaBoost
- Ann

Each model was tuned and evaluated using 3 repeats of 10-fold cross validation, a common configuration on data mining for comparing different models [10]. The model with the best scores is then chosen to make predictions in new data, defined as test set.

A random number seed is defined before the train of each one of the algorithms to ensure that they all get the same data partitions and repeats.

4 Results

In order to acquire the most adequate model to our problem, there is a need to explore which are the best parameters for each algorithm. To do so, we constructed graphs with the performances of different algorithm parameter combinations, with the final objective of finding trends and the sensitivity of the models.

The models were trained using 89159 entries, 10 predictors and 2 classes.

To guarantee that the models can be compared among themselves, there is a need to conduct hypothesis testing. A model with only the target variable and without any independent input variables can be called the null model (H0), as the model would be verified by the null hypothesis. By adding k input variables to create a fuller model (H1), we then can understand if the model is better with the input parameters or not.

We can define our H0 and H1 as such:

- **H1:** A customer's call pattern gives away its decision on churning.
- **H0:** A customer's call pattern its not connected with its decision on churning.

If we can not statistically prove H1, we must accept H0.

The p-value is the probability of observing data at least as favorable to the alternative hypothesis as our current data, considering that the null hypothesis(H0) is true. If the p-value is lower than 5%, it is very unlikely (with ¡ 5% probability) that the null hypothesis is actually true when we rejected it. So, when the p-value crosses the 5% threshold, we can reject the null hypothesis (H0) in favor or the alternative hypothesis (H1). All the models conquered values under the predefined threshold of 0.05, so we can discard H0 and accept H1.

Table 2. AUC value comparison

	Min.	1st Qu.	Median	Mean	3rd	Qu.	NA's
KNN	0.9509	0.9534	0.9544	0.9550	0.9566	0.9599	0
NB	0.6866	0.6991	0.7045	0.7036	0.7078	0.7170	0
J48	0.9276	0.9327	0.9353	0.9352	0.9369	0.9446	0
RF	0.9882	0.9898	0.9902	0.9901	0.9907	0.9915	0
ADA	0.7677	0.7771	0.7812	0.7814	0.7860	0.7970	0
ANN	0.6895	0.7082	0.7128	0.7132	0.7190	0.7318	0

The AUC is a common evaluation metric for binary classification problems. It represents the area value created by the ROC curve. If a classifier is very good, the true positive rate will increase quickly and the area under the curve will be close to 1. If the classifier is similar to random guessing, the true positive rate will increase linearly with the false positive rate and the area under the curve will be around 0.5. The values regarding AUC for our trained model are on table 2. Among the studied models, three of them achieved very good AUC values: KNN, J48 and and Random Forest.

Regarding Sensitivity values displayed on table 3, the results were worse than in the previous metric. The same models that had the higher AUC values are still on the top Regarding Sensitivity values. The AdaBoost and ANN models have very poor Sensitivity values, and can be considered not a good fit to our problem.

Table 3. Sensitivity value comparison

	Min.	1st Qu.	Median	Mean	3rd Qu.	NA's
KNN	0.8710	0.8793	0.8818	0.8817	0.8854	0.8930	0
NB	0.5475	0.5651	0.5691	0.5686	0.5751	0.5861	0
J48	0.8448	0.8610	0.8650	0.8640	0.8686	0.8752	0
RF	0.8974	0.9017	0.9054	0.9050	0.9084	0.9110	0
ADA	0.4038	0.4142	0.4346	0.4330	0.4464	0.4715	0
ANN	0.3481	0.3860	0.4372	0.4272	0.4575	0.4847	0

4.1 Variable importance

We conducted a study to acquire this values for the Random Forests method, to assess which predictors had the largest impact on the model which got the best results.

The results of this study are presented on table 4. All measures of importance were scaled to have a maximum value of 100.

Variable "duration" has clearly the most importance to our model, setting the top value of 100. The next variables significant to our model are the call "direction" and a number representing the other person on the call.

Table 4. ROC curve variable importance

Variable	**Importance**
duration	100.000
direction	61.275
other_msisdin	55.965
start_cell_id	29.924
dropped_call	24.278
destination_type	24.205
voicemail	22.547
end_cell_id	12.009
call_type	4.723
other_contract_id	0.000

5 Conclusion

This research aimed to create suitable models that predict customer churn. This models needed to register high values in the defined metrics: AUC and Sensitivity. To validate the models, we chose to implement 10-folds cross validation with 3 repeats.

The dataset was too extensive to be used in the available time to complete this dissertation, so there was a need to reduce its dimension. We took into

consideration it was more important to have the full history of a few clients instead of one month history of all the clients.

The data itself was not suitable to predict upon. It had to go through multiple transformations to make it fit to train and test the models.

As the database was provided by a company with their real values, it did not contained the information regarding the target variable: if a costumer had churned or not. According to the standard in this circumstances, we set that variable according to a simple rule: if an account did not make or receive any calls during a 30 day time period, we considered that client has cease the contract. However, when analyzing this new variable we came across another problem: it was imbalanced. Only around 18% of the entries represented churning clients. If our models were trained upon this data, it could result in overfitted models that disregard the minority class. Our approach to solve this problem consisted on applying an hybrid sampling method SMOTE. Our final dataset was much more balanced than before with the churned class represented by 0.43% of all entries, leading to balanced models.

Six models were trained and tested. The algorithms themselves were chosen due to their diversity and applicability in this kind of prediction. Those algorithms were KNN, naive bayes, random forest, C4.5, AdaBoost and ANN.

Random forest model achieved the best results in both ROC and Sensitivity in all measurements. With a highest ROC value of 0.9915 and Sensitivity value 0.9110, it distinguishes itself from the other models.

KNN and J48 models come in second and third place respectively. They acquired similar results to the random forest model, being their AUC and Sensitivity values slightly lower.

The remaining algorithms, (naive bayes, ada and ann) failed to positively predict upon our data. Their ROC values were acceptable, but their Sensitivity values are considerably low. This means that this algorithms did not conceived good models according to our metrics and needs.

As the dataset used in this research represent real calls from a telecom company, this model can effectively be used to the company's welfare. The administration, using this kind of model, can predict and target the clients that are close of ceasing their contract.

6 Future Work

Although we consider the conducted study a success, we acknowledge that there is still room for improvements.

- Since our dataset timeframe is 7 months (difference between first and last entry), some of the older data can influence in a negative way the models. It would be interesting to investigate which data timeframe produce the best result and how the efficiency of the models is influenced by this timeframe.
- The dataset available to this research has a great size regarding number of entries. However, the number of available variables to predict upon are quite

limited and we had no demographic information, considered to be important to this kind of prediction. It would be of great interest to add these variables to our data and see if they improve the models. This will also open up the possibility of conducting input selection based on the model's variable importance [11], removing irrelevant variables from the dataset that can have a negative effect on some of the algorithms (e.g., knn, ann).
– All the algorithms suffered parameter tunning to identify which parameter values returned the best models. But there are many other approaches to tune the models that were not tested and could improve the final models.

References

1. Gerpott, T. J., Rams, W., and Schindler, A.: Customer retention, loyalty, and satisfaction in the German mobile cellular telecommunications market. Telecommunications Policy, 25(4):249269. (2001)
2. Kim, H. S. and Yoon, C. H.: Determinants of subscriber churn and customer loyalty in the Korean mobile telephony market. Telecommunications Policy, 28(9-12):751765. (2004)
3. Kim, M. K., Park, M. C., and Jeong, D. H.: The effects of customer satisfaction and switching barrier on customer loyalty in Korean mobile telecommunication services. Telecommunications Policy, 28(2):145159. (2004)
4. Tsai, C. F. and Lu, Y. H.: Customer churn prediction by hybrid neural networks. Expert Systems with Applications, 36(10):1254712553. (2009)
5. Yen, C. and Wang, H.-y.: Applying data mining to telecom churn. 31:515524. (2006)
6. Oentaryo, R. J., Lim, E.-p., Lo, D., Zhu, F., and Prasetyo, P. K.: Collective Churn Prediction in Social Network. (2012)
7. Radosavljevik, D., Putten, P. V. D., and Larsen, K. K.: The Impact of Experimental Setup in Prepaid Churn Prediction for Mobile Telecommunications : What to Predict , for Whom and Does the Customer Experience Matter 3(2):8099. (2010)
8. Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P.: SMOTE : Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16:321357. (2002)
9. Wu, X., Kumar, V., Ross, Q. J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z. H., Steinbach, M., Hand, D. J., and Steinberg, D.: Top 10 algorithms in data mining. Knowledge and Information Systems. (2008)
10. Kuhn, M.: Building predictive models in r using the caret package. Journal of Statistical Software. (2008)
11. Verbeke, W., Dejaeger, K., Martens, D., Hur, J., and Baesens, B.: New insights into churn prediction in the telecommunication sector: A profit driven data mining approach. European Journal of Operational Research, 218(1):211229. (2012)