UNIVERSIDADE BEIRA INTERIOR
Engenharia

Churn Rate Prediction in Telecommunications Companies

Ana Lúcia de Morais Lima

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática
(2º ciclo de estudos)

Orientador: Prof. Doutor Pedro Ricardo Morais Inácio
Co-orientador: Prof. Doutor João Carlos Raposo Neves

Covilhã, novembro de 2021

Dedicatória

Dedico este trabalho ao universo todo poderoso que sempre esteve comigo e sabe de todos os meus dias de luta e glória. Dedico a minha família em especial a minha mãe Hélia Lúcia, minha avó Lúcia e minha madrinha Nilda, essas três mulheres são os meus pilares de motivação, fé e sabedoria. Dedico aos meus amigos que me ajudaram em todo o percusso. Dedico a todos os meus professores que me acompanharam desde de sempre em especial aos orientadores Doutor Pedro Inácio, Doutor João Neves e o supervisor João Santos que foram minha motivação diária, e meu apoio constante. Dedico a Universidade da Beira Interior que sempre me deu todo o suporte necessário através de bolsas e apoios sociais, que sempre buscou me acolher e me fazer se sentir em casa. Dedico ao departamento de contabilidade no qual desenvolvi atividades do FAS que me ensinaram outros aspectos do que é ter compromisso com o próximo. Dedico a TimWelab Tech que possibilitou que a pesquisa e desenvolvimento fosse concretizado. Dedico a todas as cozinheiras da Cantina das Engenharias na qual desenvolvi atividades do FAS, elas me ensinaram e me mostraram o que é ter disciplina e garra. Dedico a todas as pessoas da comunidade de Santa Luzia.

Agradecimentos

Agradeço ao universo por ter me mostrado a oportunidade desta bolsa de investigação que mudou a minha vida. Agradeço a toda a minha família que sempre se fizeram presente apesar de três anos distantes. Agradeço ao professor Doutor Pedro Inácio que é uma das minhas maiores inspirações como profissional e como pessoa e que possui uma essência humana muito especial. Agradeço ao professor Doutor João Neves por todo o conhecimento que me transmitiu neste percurso, pela sua paciência, compreensão, profissionalismo excepcional e dedicação em me ajudar a fazer um bom trabalho. Agradeço todos os meus amigos em especial minhas amigas Laura e Mariane por todo o apoio, força e motivação que me deram para concluir este trabalho. Agradeço ao professor Doutor Hugo Proença que me transmitiu conhecimentos essenciais para a concretização deste projeto. Agradeço ao meu supervisor João Santos que confiou plenamente no meu trabalho e que me ajudou em vários momentos. Agradeço a toda equipe TimweLab Tech, em especial ao Isento pelo excelente trabalho que fizemos em equipa na finalização do projeto Telco Vista. Agradeço a Universidade da Beira Interior que me acolheu e me abrigou nos meus momentos mais criticos. Agradeço a toda a comunidade acadêmica que fazem a diferença no mundo, compartilhando conhecimento e inspirando a evolução humana.

Resumo alargado

O setor de telecomunicações é visto atualmente como um dos setores que mais cresce no mundo, com um desenvolvimento exponencial nos últimos anos afetando cerca de 90% da população em geral [BGM+20a]. Este crescimento tem sido alimentado pelos recentes avanços tecnológicos e novos serviços de telecomunicações, implicando diretamente no aumento dos dados que se tornaram um ativo de primeira classe para empresas, corporações e organizações. Apesar do vasto número de clientes, existem múltiplas empresas operando neste mercado oferecendo serviços similares a uma gama restrita de preços. Este fator junto com os custos reduzidos de mudança entre empresas justifica porque o setor de telecomunicações é um mercado tão competitivo, onde a rotatividade de clientes é uma preocupação central para as receitas das empresas. Contudo a taxa de churn pode ser visto como termômetro para a saúde da empresa.

Uma forte concorrência entre empresas rivais e tarifas competitivas de múltiplos fornecedores são as principais razões para os clientes mudarem entre as operadoras de telecomunicações. Entretanto, outros fatores podem levar os clientes à rotatividade, tais como o aumento dos valores dos planos, atendimento deficiente ao cliente, tempos de conexão lentos, e-mails de marketing indesejados, e outros. Com base nestes fatores, a chave para mitigar este problema é prever os clientes que estão em risco de churn, ou em outras palavras, rotatividade. Ultimamente, muitos pesquisadores estão interessados em trabalhar várias técnicas para prever a rotatividade dos clientes de telecomunicações. A indústria de telecomunicações tem lutado com a ameaça de perder mais de 25% de seus clientes a cada ano, o que se acredita resultar em uma enorme perda de receita. Outro fator relevante é que adquirir um novo cliente custa entre 5 e 10 vezes mais do que manter um cliente com a empresa. Com base nisto, é essencial manter os assinantes existentes ou evitar a rotatividade dos clientes [MTMM13]. De acordo com Kortler, a redução da taxa de rotatividade em 5% aumenta o lucro de 25% para 85% para as empresas de [K+97].

Assim, tem havido uma demanda crescente para automatizar os processos utilizados e identificar a rotatividade dos clientes. Entretanto, este processo é tão caro que normalmente apenas 15% da receita obtida pelas empresas móveis é gasta em infra-estrutura de rede e TI, enquanto 15 a 20% da receita é usada na aquisição de clientes. Os modelos de rotatividade de clientes visam identificar os primeiros sinais de rotatividade e tentar prever os clientes que saem voluntariamente. Portanto, muitas empresas percebem que seus sistemas de banco de dados existentes são um de seus ativos mais valiosos e, de acordo com Abbasdimehr, [AST11] os dados internos que as empresas têm sobre seus clientes são uma ferramenta útil para prever clientes em risco.

O problema é caracterizado da seguinte forma churn é calculado dividindo o número total de clientes pelo número total de clientes ativos em um determinado período. A rotatividade de clientes pode ser gerenciada de forma reativa ou pró-ativa. Na abordagem reativa, a empresa espera o pedido de cancelamento do cliente e depois oferece planos de retenção atraentes. Na abordagem pró-ativa, a probabilidade de rotatividade é prevista de acordo com os planos oferecidos aos clientes [Pen09].

No segundo caso, as abordagens baseadas no aprendizado de máquinas provaram ser altamente eficientes na estimativa da probabilidade de rotatividade do cliente[UI16, VDSC15, AJA19]. Alguns algoritmos usados nestas estratégias são regressão linear, SVM, árvores de decisão, floresta aleatória, e Naive Bayes.
Ao construir uma estratégia baseada na aprendizagem da máquina, a análise e processamento de dados desempenha um papel significativo na melhoria da precisão da classificação. Muitas abordagens foram desenvolvidas por pesquisadores a fim de selecionar características que são úteis na redução da dimensionalidade dos dados, complexidade computacional e sobreajustes. Na previsão do churn, as características com maior grau de importância são extraídas do vetor de entrada, pois são úteis para prever os clientes que deixarão a empresa.
A fim de resolver o problema acima, as seguintes técnicas de aprendizagem de máquina foram utilizadas neste trabalho: (1) Regressão logística, (2) Naive Bayes, (3) máquina vetorial de suporte, (4) Classificador floresta aleatória, (5) Decision Tree, (6) KNN, (7) e algoritmos de gradient boosting tais como AdaBoost, XGBoost, LGBM Classifier e CatBoost. O objetivo é fazer uma análise comparativa entre estes algoritmos para prever vários padrões de rotatividade dos clientes. Além disso, para uma melhor compreensão do conjunto de dados, os dados foram pré-processados para encontrar insights importantes e vetores de características. Depois de implementados os modelos são testados em mais dois datasets que servem como uma forma de avaliar melhor seu desempenho em dados desconhecidos.

Abstract

Customer churn is a central concern for companies operating in industries with low switching costs. Among all industries, the one that suffers most from this problem is the telecommunications sector, with an annual churn rate of approximately 30%. As operators grow, so does the volume of data, and understanding and interpreting this data is necessary for operators to understand why customer churn is happening. Through data science, machine learning, and artificial intelligence techniques, the possibilities of predicting customer churn have increased significantly. In this research, the proposed methodology consists of six phases. In its first phases, data preprocessing and feature analysis are performed. In the third phase, feature selection is performed. Then, the data were divided into two parts of training and testing, in the proportion of 80% and 20%, respectively. For the prediction process, the most popular prediction models were applied, i.e. logistic regression, vector machine, naive bays, random forest, decision trees, etc. In the training set, boosting and ensemble techniques were applied to achieve better model accuracy. In the training set, K-fold cross-validation was used to avoid overlapping models. The results are evaluated using the confusion matrix and the AUC curve. The Adaboost, Catboost and XGBoost classifiers obtained the highest accuracy in the range of 85% and 92%. The highest AUC score was 98% obtained by Random Forest and 93% XGBoost which outperformed the other models.

Keywords

Data Science, Machine Learning, Churn rate, predictive model

Contents

List of Figures

List of Tables

Acronyms List

AB	AdaBoost
AUC	Area Under the Curve
CM	Confusion Matrix
CCP	Customer Churn Prediction
CV	Cross Validation
CRM	Customer Relationship Management
EDA	Exploratory Data Analysis
FAS	Fundo de Apoio Social
FN	False Negative
FP	False Positive
GB	Gradient Boostings
KNN	K-Nearest Neighbours
LR	Logistic Regression
LGBM	Light Gradient Boosting Machine
MSE	Mean Squared Erro
ML	Machine Learning
NB	Naive Bayes
RF	Random Forest
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
UBI	Universidade da Beira Interior
XGB	Xtreme Gradient Boosting

Chapter 1

Introduction

The telecommunications sector is currently seen as one of the fastest-growing sectors in the world, with an exponential development in recent years affecting about 90% of the general population [BGM+20a]. This growth has been fueled by recent technological advances and new telecommunications services, directly implying the increase in data that has become a first-class asset for companies, corporations and organizations. Despite the vast number of clients, there are multiple companies operating in this market offering similar services at a tight range of prices. This fact together with the reduced costs of switching between companies justifies why the telecommunication sector is such a competitive market, where customer churn is a central concern for companies revenues.
Strong competition between rival companies and competitive rates from multiple providers are the main reasons for customers to switch between telecom operators. Nevertheless, other factors can lead customers to churn such as increasing plan values, poor customer service, slow connection times, unwanted marketing emails, and others. Based on these factors, the key to mitigating this problem is to predict customers who are at risk of churn, or in other words, churn. Lately, many researchers are interested in working on various techniques to predict telecom customer churn. The telecom industry has been struggling with the threat of losing more than 25% of its customers every year, which is believed to result in a huge loss of revenue [WC02]. Another relevant factor is that acquiring a new customer costs between 5 and 10 times more than keeping a customer with the company [LP03]. Based on this, it is essential to keep existing subscribers or avoid customer churn [MTMM13]. According to Kortler, reducing the turnover rate by 5% increases profit from 25% to 85% for companies [K+97].
Thus, there has been a growing demand to automate the processes used to identify customer churn. However, this process is so expensive that typically only 15% of revenue earned by mobile companies is spent on network and IT infrastructure, while 15 to 20% of revenue is used on customer acquisition. Customer churn models aim to identify early signs of churn and try to predict customers who leave voluntarily. Therefore, many companies realize that their existing database systems are one of their most valuable assets, and according to Abbasdimehr, [AST11] the internal data that companies have about their customers is a useful tool for predicting customers at risk.

1.1 Problem Description

Churn is calculated by dividing the total number of customers by the total number of active customers in a given period. Customer churn can be managed in a reactive or proactive way. In the reactive approach, the company waits for the customer's cancellation request,

and then offers them attractive retention plans. In the proactive approach, the probability of churn is predicted according to the plans offered to customers [Pen09]: Churn rate = Lost customers/Total customers x 100.

$$\textbf{Churn rate} = \frac{\textbf{Lost customers}}{\textbf{Total customers}} \cdot \textbf{100}$$

By building a proactive machine learning based strategy, data processing and analysis plays a significant role in improving classification accuracy. Many approaches have been developed by researchers in order to select features useful in reducing data dimensionality, computational complexity and overfitting. In churn prediction, the features with the highest degree of importance are extracted from the input vector as they are useful in predicting the customers who will leave the company.
In this research three different types of datasets were used and the following learning techniques were applied: (1) Logistic regression, (2) Naive Bayes, (3) support vector machine, (4) Random forest classifier,(5) KNN, (6) Decision tree (7) and boosting algorithms such as AdaBoost, XGBoost, LGBM Classifier and CatBoost. The aim is to do a comparative analysis between these algorithms to predict various customer churn patterns on different types of datasets. In addition, for a better understanding of how the problem was solved, some questions are answered in order to clarify specific points of this research.

1.1.1 Research Objectives

1. Explore a telecom company's customer churn prediction using 10 machine learning classification models on three different types of datasets;

2. Investigate the impact that the turnover rate solution can have on the telecommunications industry as a whole;

3. Discuss the importance of customer churn models in the telecommunications industry;

4. Compare which algorithms and which types of datasets are effective in reducing customer churn rates in telecom companies.

1.2 Research Questions

1. What is the most appropriate machine learning model to use to predict future customer churn?

2. Which features can be considered the most important to build a predictive customer churn model?
3. What are the possible solutions to deal with unbalanced classes?
4. What impacts can the predictive model have on the telecommunications industry?

1.3 Research Limitations

1. The current study is limited to the telecommunications industry only.
2. The study does not use techniques other than machine learning techniques.

1.4 Main Contributions

After completing the development of this project, we have the following contributions:

- Development of churn prediction models using ten (10) different types of Machine Learning algorithms;
- The models developed are not limited to the telecommunications sector, but can also be applied and adapted to other areas of industry;
- An extensive state-of-the-art study on different strategies to solve the churn prediction problem;
- Integration of the proposed model in the Telco Vista project of the TimweLab company.

1.5 Dissertation Outline

This dissertation project is divided into six chapters:

1. Chapter 1 - **Introduction** - Introduces churn prediction and how telecom operators are affected. It addresses on how the problem can be solved through Data Science and ML, exposes the objectives, limitations and contributions of the project and the organization of the paper.
2. Chapter 2 - **Backgound and Related Works** - A summary is made of all the data science and ML concepts that are important for building a model. A brief description of how each model works and an overview on related works.
3. Chapter 3 - **Proposed Method for Developing the Churn Prediction Model** - It presents the methodology used to perform the data processing, information about the dataset features and libraries used. It contains detailed descriptions of the model evaluation module.

4. Chapter 4 - **Building the Churn Rate Model** - It is dedicated to explaining the model proposal, which extra datasets will be used, technologies and libraries and a brief description about each hyperparameter used for model fitting.

5. Chapter 5 - **Results and Discussion** - It sets out the results that were found by each model, each evaluation metric used and what scores were obtained, discussion of the results.

6. Chapter 6 - **Conclusion** - It presents the conclusion of the project, contributions and achievements that the project has generated.

Chapter 2

Background and Related Works

2.1 Introduction

This chapter details the main concepts and related work for understanding and developing a predictive model of turnover rate. All topics have been covered in a systematic order so that there is a better understanding of how the problem is modeled and what factors are taken into consideration.

1. 2.2 - **Background** - All the definitions necessary for a better understanding of this work are exposed.
2. 2.9 - **Related Work** - Analyze the different types of strategies that were developed to solve the problem.
3. 2.10 - **Conclusion** - Summary of all the information that was covered in this chapter.

2.2 Background

2.2.1 Machine Learning

What exactly does it mean to learn? With a brief answer we can say that learning is the ability to change with the external stimuli and also remember the experiences we have acquired in the past. Making a parallel, we observe that ML performs precisely this process that is able to mimic and even excel over human intelligence, because this powerful technology is able to learn from the environment around it. Each ML technique aims to study, design and improve mathematical models that can be trained once or continuously with data related to a given context.
With the computational power increase and the amount of data that is generated every day, ML stands out in several areas and arouses many researchers interest. Those data are successfully applied in various fields, from pattern recognition, computer vision, spacecraft engineering, finance, entertainment, computational biology and even biomedical and medical applications. In the second half of the 20th century, machine learning thrived as a subfield of artificial intelligence that involves the development of self-learning algorithms intending to offer a more efficient way to capture existing knowledge in data and thus gradually improve the performance of predictive models [Ras15].
ML is a subarea of Computer Science that is concerned with the construction of algorithms that, to be useful, depend on a set of some phenomenon examples. These examples can come from nature, be handmade by humans, or generated from another algorithm. The

term ML can be defined as computational methods that aim to use experience to improve performance or to make accurate predictions. In this case, experience is related to information about the past, which is often electronic data, whose size and quality are of great importance for the predictions rate of success that the algorithms will make. ML seeks to solve a practical problem by gathering a set of data and thereby constructing a statistical model through algorithms [Bur19].

2.2.1.1 The three different types of Machine Learning

The taxonomy or way of organizing ML algorithms can be divided into three different learning styles, as following: supervised learning, unsupervised learning and reinforcement learning. In Figure 2.1 the three different types of learning are presented.

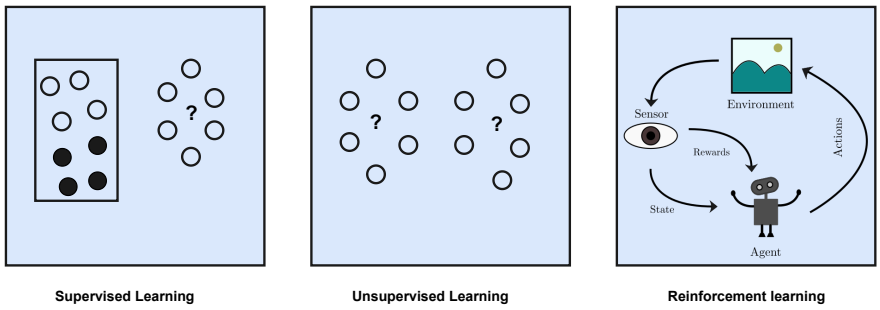


Figure 2.1: Types of Machine Learning.

2.2.1.2 Supervised Learning

This type of learning tries to predict a dependent variable from a list of independent variables. One of its basic characteristics is that the data used to carry out the training contains the desired response, that is, it contains the dependent variable resulting from the observed independent variables [Bur19].

2.2.1.3 Unsupervised Learning

This type of problem only operates with input data, with no outputs or target variables. In general, unsupervised learning seeks to find a more informative representation of the data, condensing information into more relevant points and density estimation that involves summarizing the data distribution [Bur19].

2.2.1.4 Reinforcement Learning

In this type of learning, the machine tries to learn the best action to be taken. The agent learns to achieve a goal in an uncertain and potentially complex environment, an approach that takes this uncertainty into account is desirable, and thus be able to incorporate any changes in the environment of the best decision-making process. Basically the computer uses trial and error to find a solution to the problem. In this process, artificial intelligence

receives rewards or penalties for the actions it performs. The goal is to maximize the total reward [SB18].

2.3 Feature Scaling

There are different approaches to rescale values to a desired range. One of the most common ways for data normalization is standard scale. This technique assumes that the data is normally distributed within each feature and scales it so that its distribution has a mean value of zero and a standard deviation of one. The mean and standard deviation are calculated for the feature and then the feature is scaled based na seguinte formula. Onde μ = mean and σ = Standard Deviation.

$$x_{new} = \frac{x - \mu}{\sigma} \tag{2.1}$$

2.4 Label Encoding

In machine learning, one commonly deals with data sets that contain multiple labels in one or more columns. These labels can be in the form of words or numbers. Label encoding is the process of transforming non-numeric values into numeric values. This is done because ML models require all input and output variables to be numeric. Machine learning algorithms decide the best way these labels are to be operated. So, this is an important step for the structured dataset in supervised learning [ZC18].

2.4.1 One Hot Encoding

For categorical-type variables, between which there is no ordinal relationship, the entire coding may not be enough, and causes the model to get misleading results with poor performance. The algorithm takes the variable with the nominal values and divides the column describing the attribute into multiple columns, assigning a binary value of 1 (one) or 0 (zero) to those columns. The number of divisions depends on the cardinality characteristic. One hot encoding makes training data more useful and expressive and can be easily resized [ZC18].

2.5 Overfitting and Underfitting

Overfitting is related to a model that fits very well to training data. Overfitting occurs when a model learns the details and noise in the training data, to the extent that it negatively affects the model's performance with new data. Noises and fluctuations in data become concepts learned by the model. The problem is that these concepts fail to apply to new data and negatively impact the model's ability to generalize. Underfitting occurs when

the model cannot reach a certain complexity and it is not able to model the training data or generalize the new data [Lau20].
In Figure 2.2 there is a demonstration of how this problem is characterized in practice.

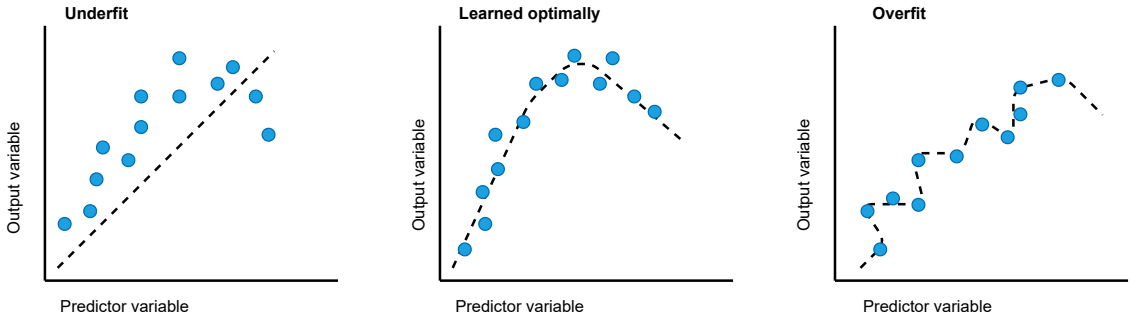


Figure 2.2: Showing the phenomena of underfit and overfit.

2.6 Data Preprocessing and Model Optimization

Data pre-processing is one of the fundamental parts in the process of creating an ML model. In this step, a set of data mining techniques is applied that involves the transformation of raw data into an understandable format. Pre-processing has a considerable impact on the model generalization performance and on improving the model understandability. This technique encompasses removing inconsistencies and converting raw data into meaningful information that can be efficiently managed. It is extremely important to remove null or missing values from the dataset and check them against unbalanced class distributions, which is a problem inherent in data extraction [GFH09].

2.6.1 Data Cleaning, Nomalization, and Transformation

This phase is crucial to reduce data dimensionality. As the dimension increases, more time and computational power are required. There are two approaches that can be implemented: filtering and packaging. Filtering is related to noise removal (outliers), words with orthographic errors, duplicate values or nonlogical data (ex: client with 100 years of loyalty time). Packaging focuses more on data qualification, by detecting and removing incorrect labels [VDA16].
The goal of normalization is to transform the resources on a similar scale. This step improves the training model performance and stability. It is essential for many machine learning algorithms such as KNN to avoid conflicting values in relation to values that are on different scales [AH01].
Transformation or feature construction is a process by which a set of new features is created. Through feature extraction it is possible to create new feature variations. Assuming the original set consists of $B_1, B_2, ...B_n$ features, these variants can be defined below.
Feature construction is the process of discovering missing information about the relationship between features by creating additional features [Tho92, LM98, WM94]. After

feature construction,there may be additional m features $B_{n+1}, B_{n+2}, ...B_{n+m}$.

2.6.2 Missing Data

Often the data that is used to create an ML model has missing values, this is a reflection of the confusion of real world data. Missing data is one of the most common sources of code errors and the reason for most exception handling. One method to fix this problem is to delete the instance that contains missing data, which often leads to data loss, and also a reduction in the amount of available data. Missing values might be filled with some estimated value. These values can be derived from similar cases using statistical methods or machine learning [ZHL12].

2.6.3 Sampling

Often in the CCP problem, there is a phenomenon called class imbalance. When an ML classifier is used with an imbalanced class distribution, failures such as inappropriate rating metrics, missing data, data fragmentation, low generalization rate and deceptively optimistic performance can occur. This is because many ML algorithms rely on the distribution of classes from the training dataset to assess the probability of observing examples in each class when the model is used to make predictions [PXBJ13]. Data sampling has a set of techniques with approaches aimed at specific problems when it comes to imbalanced data. Sampling and resampling methods can handle imbalanced learning directly because they are simple to be understood and implemented, and once applied to transform the training dataset, a set of standard ML algorithms can be used directly [PXBJ13].

2.6.4 Feature and Variable Selection

According to Ockham's Razor principles, the less complex an ML model, the more likely that a good empirical result is not just due to the sample peculiarities [RF99]. Based on this principle, feature selection aims to reduce the model complexity. The goal is for the model to be straightforward and economical in its calculations, with very little or no degradation in predictive accuracy. In building an ML model it's vital to incorporate solely the foremost important and helpful variables, as this can be a technique to cut back model complexity. Feature choice has three different kinds of strategies *Filterin*, *Wrapper methods* and *Embedded methods* [ZC18].

In Figure 2.3 below, the hierarchy of feature selection techniques is presented.

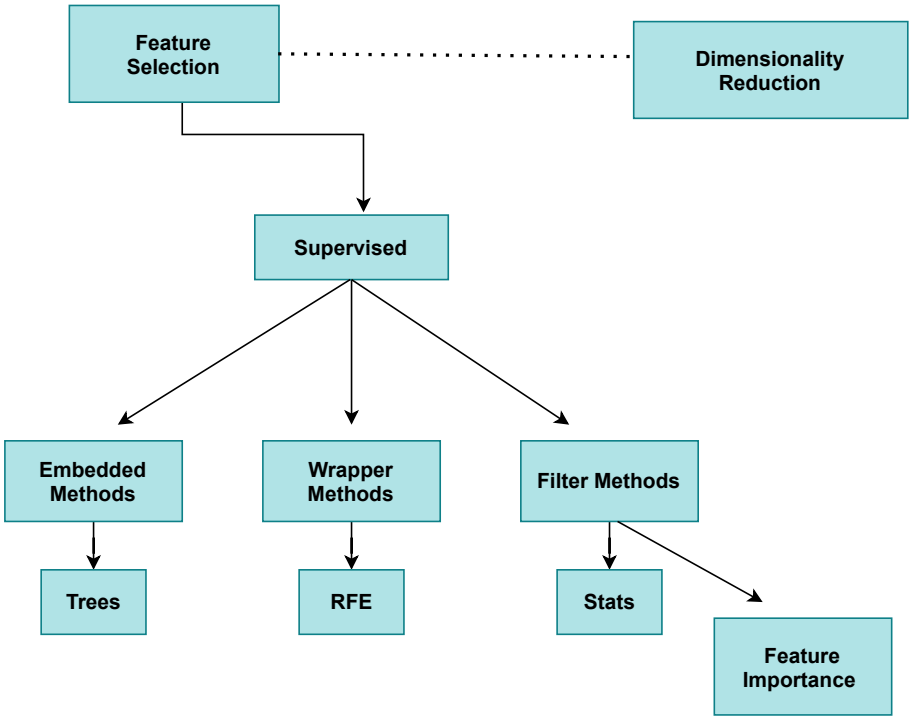


Figure 2.3: Diagram of Feature Selection Techniques.

- *Filtering* techniques are part of preprocessing the data to remove features that are unlikely to be useful to the model, and are usually the first step in any feature selection pipeline. Most filter methods calculate a score for all features and then select the features with the highest scores. In this technique it is common to use statistical measures of the correlation type between input and output variables [ZC18].

- *Wrapper methods* use algorithms to go through possible subsets of features and thus seek to maximize classification performance. This method follows a greedy research approach, evaluating all possible combinations of features against the evaluation criteria. This criterion is the performance measure that depends on the type of problem to be solved. A practical example of this method is (RFE), which selects features by recursively considering smaller and smaller sets of features. This is an efficient approach for eliminating features from a training dataset for the feature selection [KJ+13].

- *Embedded methods* combine the qualities of filter and wrapper methods. They are implemented by the algorithms that have integrated feature selection techniques. Examples of this method are tree-based models such as decision tree and Random Forest, which are well-established algorithms that already perform this process internally. This method has some advantages in being faster because it reclassifies different subsets and incorporates feature selection into the training process and it is much less subject to excessive tweaking [ZC18].

2.6.5 Hyperparameter Optimization

Many ML models have parameters that can be tuned before training starts, such as size of the hashing space, number of decisions trees and their depth, kernel, etc. These parameters are known as hyperparameters and if well used can make ML models get an effective performance based on a chosen criterion, such as accuracy or recall rate. Hyperparameter searches can be done manually or automatically. Automatic search outperforms manual search as it has benefits such as speed and reproducibility [CDM15]. Defining the optimal values for hyperparameters can be a challenging and resource-intensive task. So there are some strategies to automatically optimize hyperparameters, which are: grid search, random search, Bayesian optimization, gradient based optimization, and others. Grid search is an adjustment technique that intends to calculate the ideal values of hyperparameters. In this technique, an exhaustive search is carried out on all possible parameter values specific to a model. Furthermore, this technique is able to save time, effort and resources [FH19]. Another well-known way to perform the optimization is random search. This is a technique that makes random combinations of hyperparameters to find the best solution for the built model. Despite being similar to grid search, comparatively random search is able to produce better results. A disadvantage of this technique is the large variation in the computation of results, as the selection of parameters is completely random and consequently there may be some hyperparameter that affects the performance of the classification model [BB12].

2.7 Data Mining and Machine Learning Mechanisms in the Operators' Business

Daily, a large amount of data is generated and processed by telecom operators' systems. Much of this data can generate relevant knowledge for monitoring operational indicators of the operators' business, often using data mining and machine learning mechanisms to generate relevant information from this data and identify important patterns for the business. As described in [Wei05], there are three fundamental types of data for telecommunications operators that can generate important knowledge for them: data related to call detail records, network performance, and consumer information. However, acquiring knowledge through this data is not always easy, since: (i) operator databases can contain billions of records; (ii) there are events that rarely occur and are difficult to predict, such as fraud or network failures; (iii) many of the events associated with operators that need to be detected (fraud, again) occur in real time, making it difficult to apply Data Mining mechanisms; and, (iv) some of the data acquired is not initially found prepared for mining, requiring pre-processing. Data mining mechanisms are also used to improve sales and marketing services in the telecommunications field, as shown in [EOUD17]. In this work, the authors used data related to sales of products or services rendered by Nigerian operators between the years 2008 and 2015, and the sales of airtime credit services, balance recharges and SIM card sales during this period were analyzed. In order to predict the sales rate of each product/service in the following years (23 years). However, and despite

presenting practical results for the period between 2008 and 2015, sales forecasts were not very easy to be identified. Churn prediction in telecommunications operators is explored in other works found in the academic literature, such as in works [MA16, YRS, QRQ+13]. In these works, a study was made about the most adequate data mining and machine learning algorithms to predict the churn of operators' customers: in [MA16] the prediction models were developed using the CN2 algorithms, decision trees and Naïve Bayes, obtaining a better precision for the CN2 algorithm; in [YRS], the algorithms KNN, Naïve Bayes, Random Forest, AdaBoost and ANN were used to obtain a better precision for the churn calculation through the Random Forest algorithm; in [QRQ+13], a study of some machine learning algorithms to calculate the churn of operators' customers was demonstrated, namely through linear regressions, logistic regressions, artificial neural networks, kmeans clustering, and decision trees. Data from 106,000 customers were used over 3 months, with the best results obtained through decision trees (Exhaustive CHAID algorithm). In this work, a data resampling process was used to solve the unbalanced classes problem (large percentage of non churners vs small percentage of churners) which led to an ineffective calculation of the probability of a client being able or not to become a churner. However, two issues could be highlighted after analyzing these works: the effectiveness or the data mining and machine learning algorithms accuracy degree to be used to predict the churn of carrier customers varies depending on the number of records to be considered and the attributes chosen; and most studies carried out on the churn of operators' customers only consider the possible abandonment of customers, not considering the reasons for this to happen. Data mining and machine learning mechanisms can also be used to detect fraud from carrier customers. An example is given in [KNN16], where a fraud detection system for the telecommunications sector using artificial neural networks was conceptualized. More precisely, this work was focused on detecting fraud in subscription to services by analyzing different subscribers information, such as name, age, gender, registration period, subscription type and amount of mobile data used. For this purpose, it was necessary to collect data, pre-process them, build the neural network, train the model, and test its performance. In the end, an accuracy of around 85% was obtained. However, the number of records considered was relatively low (1000) and the customer indicators used were limited. In [Hil12] a fraud detection technique in the telecommunications area was described using data mining mechanisms exclusively on customer behavioral data, not invading their privacy when accessing their personal data (eg location). Neural networks, decision trees and agglomerative clustering were used, and a better performance was obtained with the use of neural networks. However, they did not allow the discriminative characteristics to be accessed. Additionally, the author of this work emphasized that the use of information from social networks could be useful for the definition of different customer profiles. This consumer profiling turned out to be important for discovering other factors that negatively influence the business, such as bad debt. In this sense, in [MC13], a model for predicting the insolvency of telecommunications operators customers was presented, based on the behavior analysis typically presented by insolvent customers before the deadline for debts payment. The authors of this work developed

this model based on the Fuzzy Logic methodology, segmenting customers into insolvent or non-insolvent depending on the result returned by the analysis of the customers' personal data and customer account data (time in call, purchased services, previously failed payments, etc). In order to classify customers into solvent or insolvent, the authors used three distinct algorithms: neural networks, decision trees, and Naïve–Bayes, having obtained similar precisions for the three, but slightly better for the case of neural networks. As it was possible to analyze in the state of the art, the data mining and machine learning mechanisms were currently used to identify fraud, predict customer abandonment (churn), analyze bad debt situations, and segment customers into different profiles at the operators. However, some questions arose, including the type of data to be used to carry out consumer profiling, as the use of personal customer data may violate some existing privacy rules. Still, the single and exclusive use of customer behavioral data proved to be insufficient to obtain accurate consumer profiling. Additionally, the identified churn prediction models did not contemplate the reason for such an event to take place, thus not providing a sufficiently logical analysis of it. Through the analysis we carried out, it was also possible to verify that there was no single system that performs consumer profiling in order to simultaneously obtain churn forecast models, bad debt forecast, loan supply (airtime credit) and product supply/custom services in a single tool, so there is room to innovate in this field.

2.8 Algorithms

This section presents brief concepts about machine learning models that can be used to solve the churn prediction classification problem.

2.8.1 Random Forest

Random Forest is a versatile model to solve classification and regression problems. The algorithm consists of a large number of individual decision trees operating as a set. The model trains several non-correlated decision trees and makes predictions using the most repeated results in case of a classification problem, or the mean of the values obtained in case of regression. This model has reduced variance which implies more consistent results, and consequently, a more robust model [Bre01].

In the Figure 2.4 below we have an illustration of how the Random Forest algorithm works.

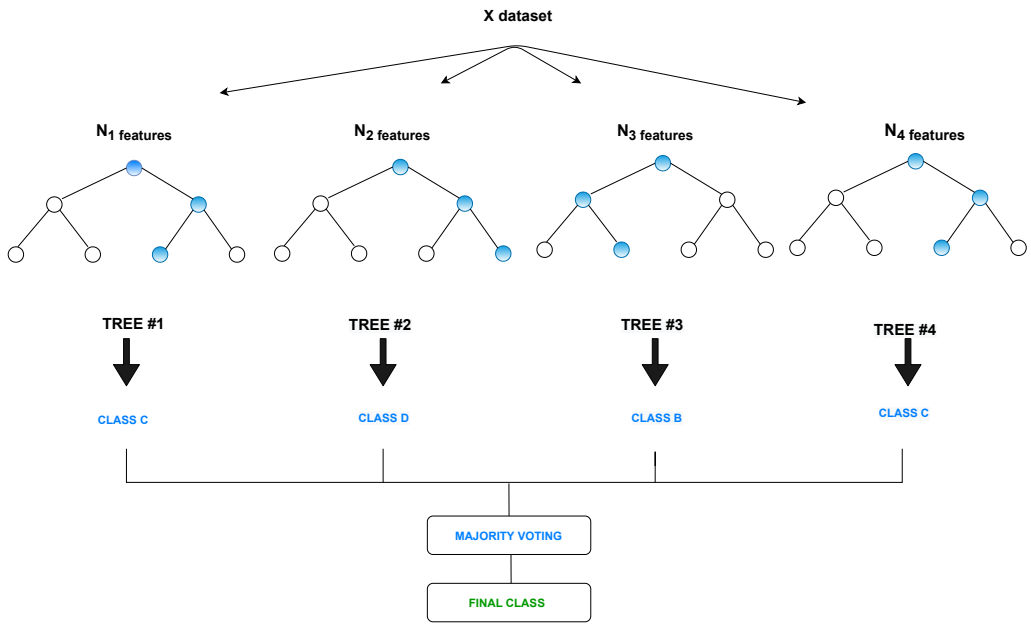


Figure 2.4: Schematic structural of the Random Forest model.

2.8.2 SVM

Support Vector Machine can be used to solve classification and regression problems. The purpose of the support vectors algorithm is to find a hyperplane in a N-dimensional space that ranks the data points distinctly. The support vector machine divides the forecast into two parts, +1, which is the right side of the hyperplane, and -1, which is the left side of the hyperplane [ZLL+05].

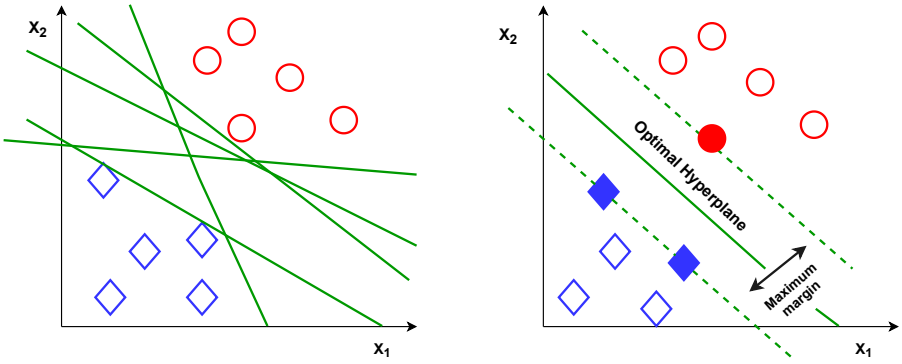


Figure 2.5: Schematic structural of the Support Vector Machine (SVM).

2.8.3 Decision Trees

Decision trees are models with great popularity, being easy to be trained and interpreted [Sab18]. The tree structure is similar to a flowchart where an internal node represents the attribute, each branch represents a decision rule, and each leaf node represents the result. The node at the top is known as the root node that will split records into two or more nodes. The ramifications that exists between nodes represents the rating rules that can be described with If-Then-Else rules. This type of model can be applied to categorical or continuous

data, and the results of their forecasts are reasonably satisfactory, but are prone to overfitting [SZ06].

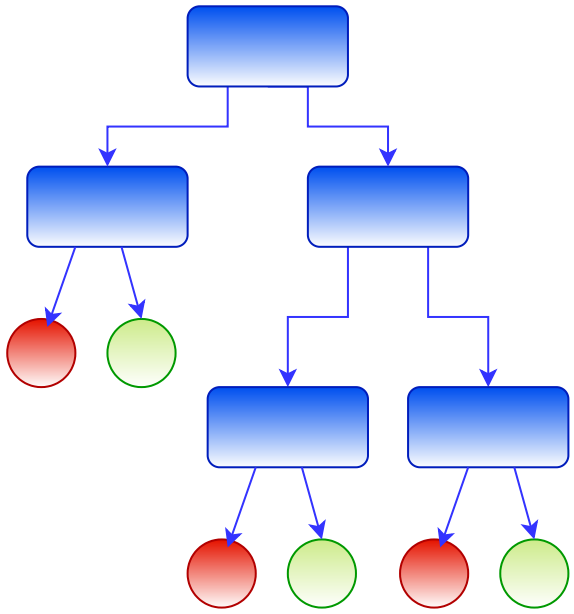


Figure 2.6: Schematic structural of the Decision Tree Classifier.

2.8.4 Logistic Regression

Logistic regression (LR) is mainly used to investigate and estimate how the dependent variables are related to the independent variable. In the logistic regression, the dependent variable has only two categories. The occurrence of an event is characterized as "1" and the absence as "0" [Sab18]. For this reason, this model is commonly used to address binary classification problems, such as the churn rate prediction [Kar98].

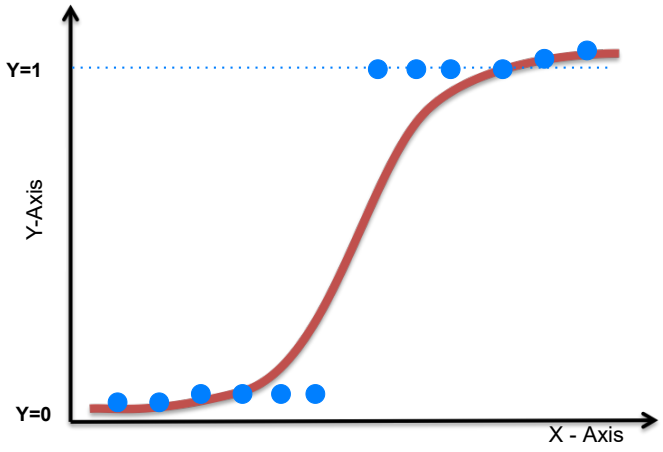


Figure 2.7: Schematic structural of the Logistic Regression Classifier.

2.8.5 K-Nearest Neighbor (KNN)

KNN is one of the most famous supervised learning algorithms and can be used to solve classification and regression problems. Its learning is information-based or memory-based, where new instances are labeled based on previous instances, and then, stored in memory [Sab18]. KNN works using the distances between the data points to sort the records, where euclidean distance is often used. This model can find several neighbors, with the number of neighbors being given by K. The minimum value of K is 1, meaning that at least one neighbor must be used for prediction. The maximum value is the number of data points that exit, which means using all neighbors. The value of K is defined at the time of implementation of the model [BGM+20b].

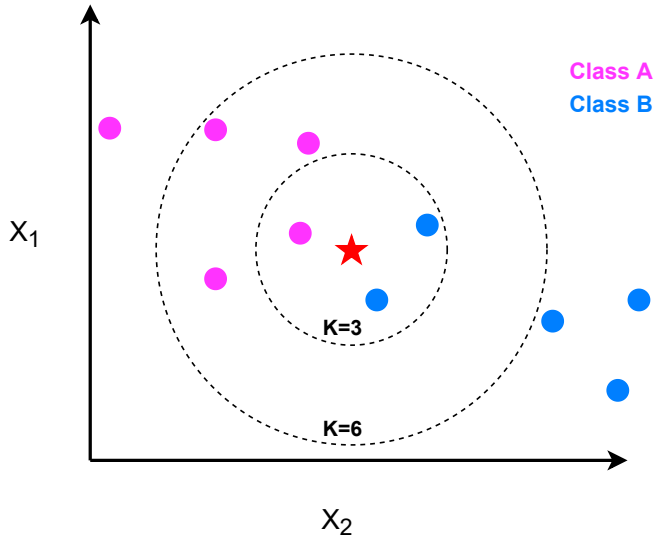


Figure 2.8: Schematic structural of the K-Nearest Neighbor (KNN).

2.8.6 XGBoost Classifier

XGBoost - Xtreme Gradient Boosting - is an implementation of decision trees with a gradient increase designed for improving speed and performance. It is considered one of the most powerful algorithms for its scalability capacity, which drives fast learning through parallel and distributed computing, offering efficient memory usage. The learning set consists of a collection of several predictors to provide better prediction accuracy. In this technique, errors committed by previous models are corrected by successive models by iteratively adapting the weights of the models [Reg21].

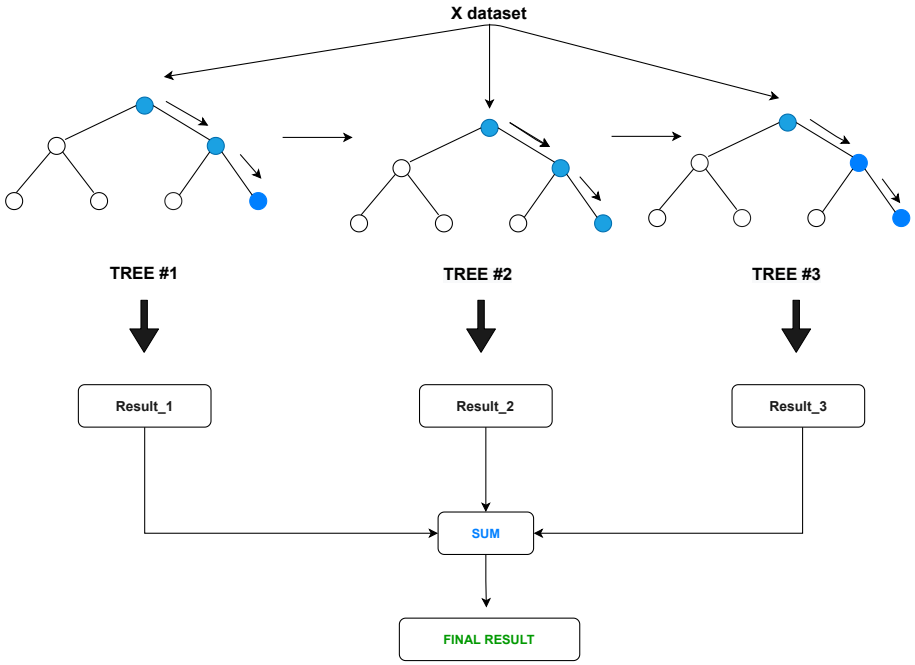


Figure 2.9: Schematic structural of the XGBoot.

2.8.7 Adaboost

AdaBoost - Adaptive Boosting - is a boosting technique that combines weak classifiers into a single strong classifier. Adaboost commonly uses decision trees as weak learners. The output of the weak learners is combined into a weighted sum representing the final output of the optimized classifier. Adaboost uses previous results and, checks their flaws and impulses adjustments in favour of the incorrectly classified instances. In short, AdaBoost reintroduces the algorithm iteratively, choosing the training set based on the accuracy of the previous training [Sch13].

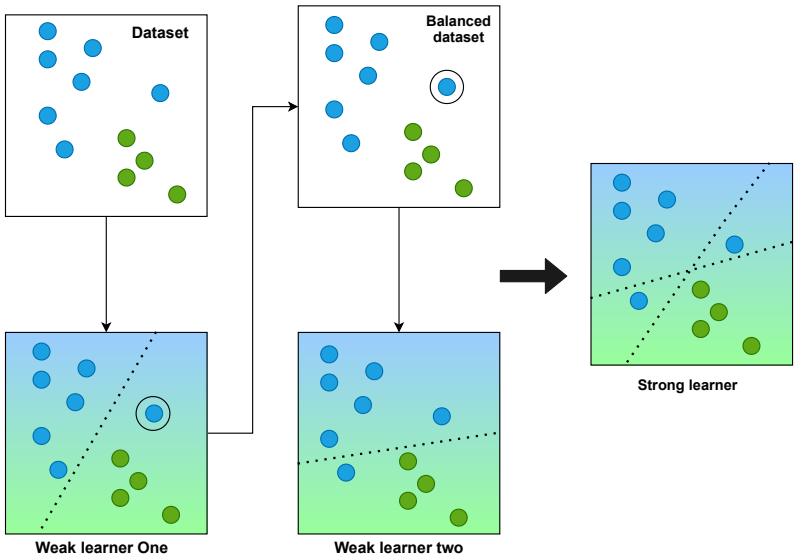


Figure 2.10: Schematic structural of the AdaBoost.

2.8.8 Catboost

CatBoost is a decision tree algorithm based on gradient increase. It is considered a simple algorithm to be implemented and it is very powerful. The model is able to provide excellent results on its first run. One of the considerable advances in CatBoost is that it already includes some of the most commonly used pre-processing methods (e.g., automatic encoding, coding labels), reducing the need of using these techniques to prepare data for training [PGV+17]. CatBoost implements symmetrical trees, which ensures a smaller prediction time.

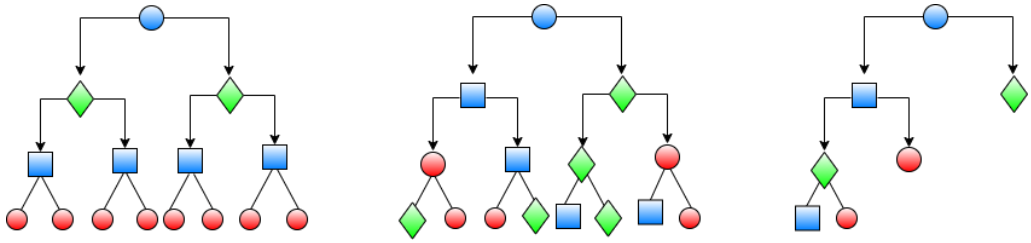


Figure 2.11: Schematic structural of the CatBoost.

2.8.9 LightGBM Classifier

One of the main goals of LightGBM is to speed up training while maintaining XGBoost-like performance. For this to happen, optimizations are performed during the tree building process. LightGBM stands out for the speed of its training, good accuracy with default parameters, low memory consumption and its ability to handle large datasets. The classifier also presents a large set of hyperparameters that are used to adjust the model. In the Figure 2.12 below we have the leaf-wise tree growth approach which is used at each level of the tree construction, only one side of the tree gets deeper. So at each level we have a smaller amount of residues to consider in order to find the threshold that maximizes the gain, in this step there is also the acceleration of the training process [sph21].

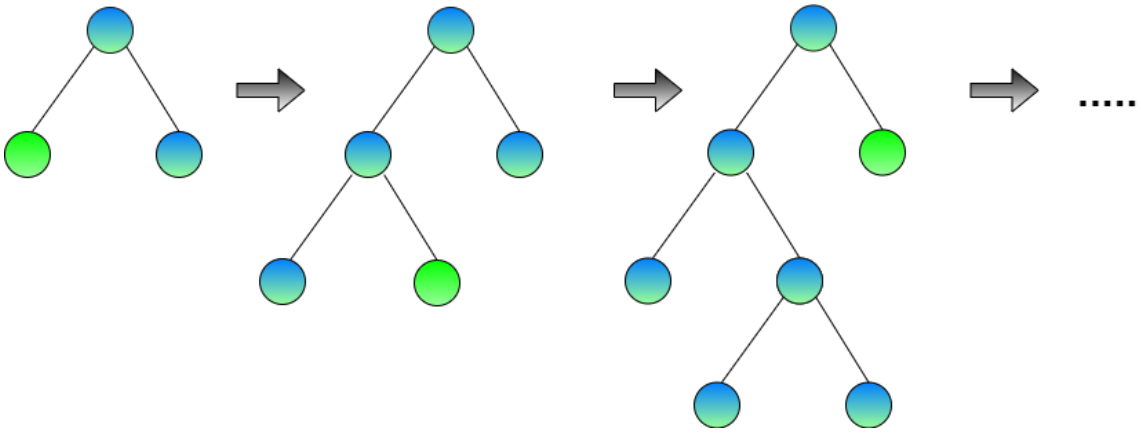


Figure 2.12: Schematic structural of the LightGBM Classifier.

2.8.10 Naive Bayes

Naive Bayes (NB) is a statistical classification technique based on the Bayes theorem. NB estimates the likelihood of an event happening based on knowledge of the variables associated with it. It is one of the simplest supervised learning algorithms. The Naive Bayes classifier is a fast, accurate and reliable algorithm. In this algorithm, the variables are assumed to be independent, since the presence/absence of a characteristic is not related to the presence/absence of any other characteristic [URM+19].

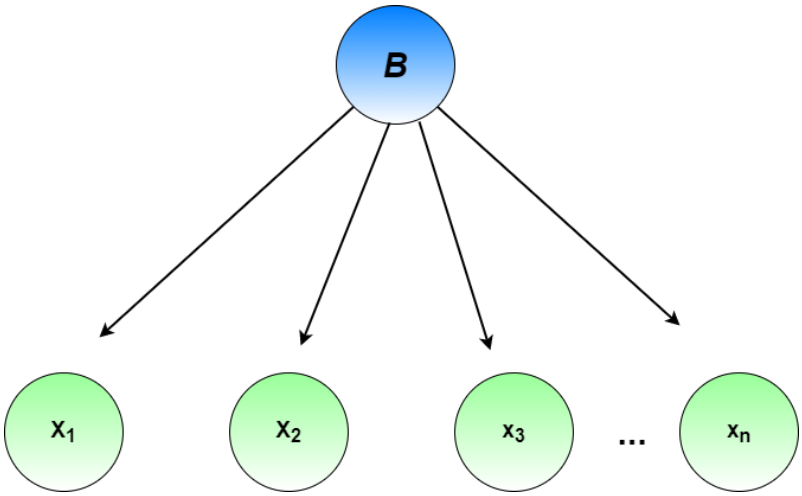


Figure 2.13: Schematic structural of the Naive Bayes Classifier.

2.9 Related Work

Churn prediction was addressed in the literature using multiple techniques, including machine learning, data mining and hybrid methodologies. These techniques help companies to identify, predict and retain customers, while assisting customer relationship management (CRM). This section describes the state-of-the-art works devised for churn prediction in the telecommunications sector.

Bhawna Nigam et al. [BN19] used deep neural networks to fit the data using various hierarchies of concepts, allowing to increase the performance of the built models. The model was trained using the K-Fold cross-validation technique. Thus, the model was able to achieve 85% of the churners that are correctly planned as churn customers [BN19]. In [AJA19] the authors applied tree-based algorithms for forecasting customer churn, namely, decision trees, random forest, generalized boosted models (GBM) tree algorithm, and XGBoost. The comparison among these algorithms showed that XGBoost model performed better than others regarding AUC accuracy. Sahar F. [Sab18] held a comparative study among ten algorithms belonging to different machine learning categories. The chosen algorithms included discriminatory analysis, decision trees, learning based on examples (K-nearest neighbors), support vector machines, logistic regression, set-based learning techniques such as Random Forest, AdaBoosting Trees and Stochastic Gradient Boosting, Naive Bayes, and multi-layer perceptron. According to the author's results, the mod-

els that obtained better results were Random Forest and AdaBoosting which presented the same accuracy (96%), and the multi-layer perceptron techniques and support vector machines with 94% precision. In [URM+19] the authors proposed a churn forecasting model that uses classification and grouping techniques to identify potential churners. The model also provides some factors behind customer churning in the telecommunications sector. The authors first classify churn customer data using common ranking algorithms. The Random Forest algorithm (RF) used correctly identified 88.63% of the instances. The churn prediction model proposed by the authors was evaluated through metrics, such as accuracy, recall, f-score and receiver operating characteristic (ROC).

Horia Beleiu et al. [BTB16] adopted three machine learning approaches for customer churn forecasting: neural networks, support vector machine and bayesian networks. On the features selection process, the principal component analysis (PCA) was considered to reduce the dimensionality of the data. However, the features selection process can be improved using an optimization algorithm that increases the classification accuracy. For performance evaluation, gain measure and ROC curve were used. Anurag Bhatnagar et al. [BS19] used two machine learning models (KNN and logistic regression). The authors used the confusion matrix to evaluate their results. By comparing the classification models, the best result was obtained by the KNN with 88.5%, followed by the logistic regression model, which obtained an 86.5% precision. In [AMR17] two different dimensionality reduction algorithms were proposed: correlation-based features (CFs) and information gain (IG). Also, three classification models were used, namely Bayesian Networks, Simple Logistic and Decision Table. Experimental results have demonstrated that the classifiers performance improved when reducing the number of client churn dataset features. [AMR17].

In [JKS20] the authors implemented two machine learning techniques that were logistic regression and LogitBoost. By observing the results, the authors concluded that there was not much difference in the results returned by both techniques, since logistic regression and LogitBoost had an accuracy of around 85%. In [LMCS21] the author uses the gravitational search algorithm as a feature selection technique. The author chooses models known in the literature and evaluates the results through the AUC curve and confusion matrix. Ensemble methods obtained satisfactory results when compared to the other models that were used. The highest AUC score was 84% and it was achieved by the AdaBoost classifier and the XGBoost classifier, which outperformed the other models. In [DCCDB18], the authors implemented a new hybrid algorithm called Logit Leaf Model (LLM) for customer churn forecasting. This new hybrid approach is compared with decision trees, logistic regression, random forest and logistic model trees with respect to predictive and comprehensibility performance. The proposed LLM model offers an understandable method with benefits in relation to the action capacity of the model, which is its main advantage over the logistics model trees and random forests.

Table 2.1 summarizes the most recent works on churn rate prediction, where it can be observed that the simplicity of a model makes all the difference when it comes to good results, despite the different Machine Learning techniques that were applied. Based on

the analysis of the different results obtained in the related work, it was possible to conclude that most authors have opted to use the most popular classification techniques of supervised learning.

Table 2.1: Summary of Literature Review

Authors	Year	What?	Techniques Used	Dataset	Evaluation and validation
Bhawna Nigam, Himanshu Dugar, Niranjanamurthy M	2019	We used the h2o package in R to build the artificial neural network and predict the turnover	H2O Deep neural network (DNN) Cross validation Regularization	Telecom churn Data set	Confusion Matrix, AUC accuracy
Ahmad, Abdelrahim K. Jafar, Assef Aljoumaa, Kadan	2019	The model developed in this work uses machine learning techniques on big data platform and build a new way of features engineering and selection.	Decision Tree, Random Forest, Gradient Boosted Machine Tree Extreme Gradient Boosting "XGBOOST"	Data provided by SyriaTel telecom company	AUC
Ullah, Irfan Raza, Basit Malik, Ahmad K. Imran, Muhammad Islam, Saif Ul Kim, Sung Won	2019	Feature selection is performed using information gain and correlation attribute classification filter. The proposed model first classifies churn customer data using classification algorithms.	Random Tree (RT), J48, Random Forest, Decision Stump, AdaboostM1 Decision Stump, Bagging Random Tree, Naïve Bayes, Multilayer Perceptron, Logistic Regression, IBK and LWL	Uses two sets of churn-bigml and South Asian telecom data	Accuracy, TP rate, Recall FP rate, precision, F-measure, ROC Area
A. Bhatnagar S. Srivastava	2019	Performance analysis between the KNN and Logistic Regression models	KNN and Logistic Regression	Data from a telecom company	Confusion Matrix, accuracy
Jain, Hemlata Khunteta, Ajay S. Sumit	2020	Perfomance analysis between the Logistic regression and Logit Boost	Logistic regression Logit Boost	Used Orange Data set	TP Rate, FP Rate, precison, Recall, accuracy F-measure, ROC Area, PRC Area

2.10 Conclusion

This chapter discussed the main concepts related to machine learning that are used in this research and also concepts that are involved in the process of building and developing a predictive model of churn rate. The relevance of using data mining and machine learning to solve problems of this type is explained in Section 2.7. A brief description was made of the classifiers commonly used in machine learning was provided, along with a graphical explation of them. In the related work was made a summary of the main related works and the different types of approaches that can be used to solve this type of problem.

Chapter 3

Proposed Method for Developing the Churn Prediction Model

3.1 Introduction

This chapter presents the methodology used to create the machine learning model which was one of the extremely important factors in conducting this study. It also presents graphs of extracted data that generated relevant insights for the realization of this model.

1. 3.2 - **Data preprocessing** - Reports all the important steps that were carried out before the model was built.

2. 3.3 - **Model evaluation** - Presents the main methods, namely Confusion Matrix, Cross-validation and AUCROC curve that are used to evaluate the models that were developed in this study.

3. 3.4 - **Conclusion** - Summary of all the points that have been covered in this chapter.

3.2 Data preprocessing

As described in section 2 data processing has a number of techniques that are used to transform and clean data in a useful and efficient manner. As well as being a powerful technique for visualising data and solving potential problems.

3.2.1 Method Overview

The pipeline of the proposed approach is illustrated in Figure 3.1. First, the data is pre-processed, including data filtering, outlier removal, data normalization, data balancing and removal of unnecessary features.
In the normalization process, a function called tenure is created, which sorts the time that each client is in the service. After performing pre-processing, the data is divided into two parts, one for training and one for testing. Then, different sorting algorithms are applied to categorize customers such as churners or non-churners. The rating algorithms used are Random Forest, Decision Tree, AdaBoost, KNN, XGBoost, Naive Bayes, LGBMClassifier, CatBoost, SVM, Logistic Regression. In this stage, it is also possible to analyze the features that are considered with a greater degree of relevance to the model.

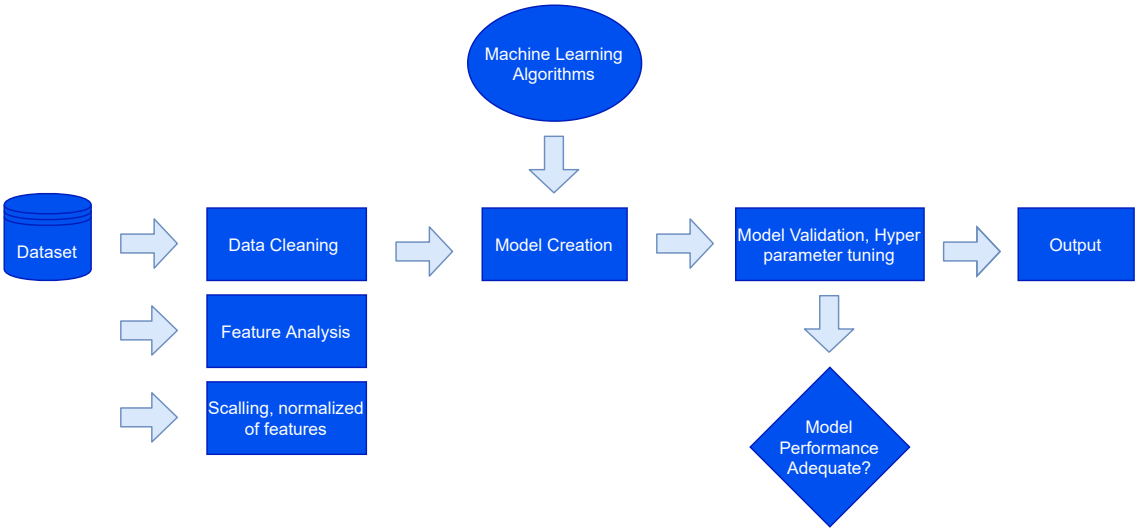


Figure 3.1: Architecture of the proposed customer churn detection approach.

3.2.2 Data description:

Understanding why customers stop purchasing a service is essential for a business to remain healthy, profitable and successful. To develop the customer churn model a dataset originally made available on the IBM Developer learning platform was used. The study classifies which customers will leave as ”1” and which will not as ”0”. The CCP can be examined based on the company’s systematic customer history. The dataset is composed of 21 features, 16 of them are categorical and 5 are numeric, as described in Table 3.1.

Table 3.1: Features and their types

Features	**Types**
Customer id	numeric
Gender	categorical
Senior citizen	numeric
Dependents	categorial
tenure	numeric
Partner	categorial
Phone Service	categorial
Multiple line	categorial
Internet service	categorial
Online security	categorial
Device protection	categorial
Tech suport	categorial
Streaming Tv	categorial
Contract	categorial
Paperless biling	categorial
Payment method	categorial
Monthly charges	numeric
Total Charges	numeric
Churn	categorial

As we can see there are features that can be explored further in order to gain significant insights that help in the understanding and construction of the models.

3.2.3 Feature Analysis - EDA

EDA consists in understanding how the data set is distributed in order to perform an in-depth investigation of it with the aim of detecting anomalies, testing some hypotheses and verifying some assumptions with the help of historiograms, graphs, among other means [VDA16]. The aim is to make more sense of the data before building the model. For the data analysis, the following libraries were used:

- **Pandas**- open source library, flexible, easy to use and with high performance power. Popularly used for data analysis for Python programming language [Pan21].

- **Numpy** - very important package for scientific computing in Python. The library provides a multidimensional array object and a variety of fast operations such as: logic, selection, arrays, basic statistical operations, random simulation and among other operations [Num21].

- **Matplotlib** - is a powerful library for making 2D matrix graphics in Python. The library is able to create simple plots with just a few commands, or just one. Great for data analysis and visualization. The tool can interact with complex data structure, interacting with database and service pains http [IT21].

- **Seaborn** - is a Python data visualization library based on Matplotlib. Its distinguishing feature is its high-level interface for designing attractive and informative statistical graphs. The tool aids in data exploration, understanding and visualization. The plotting functions operate on dataframes and matrices containing entire datasets. Its declarative and dataset-oriented API allows for a more versatile and easy-to-understand graph [Pyd].

- **Missingno** - Python library that assists in the process of identifying missing values through informative graphs [Ale21].

3.2.3.1 Distribution analysis of costumer churn dataset different attributes.

Regarding the gender turnover rate, female customers are more likely to commit turnover than male customers, but the difference is minimal, approximately $(\pm 0.8\%)$.

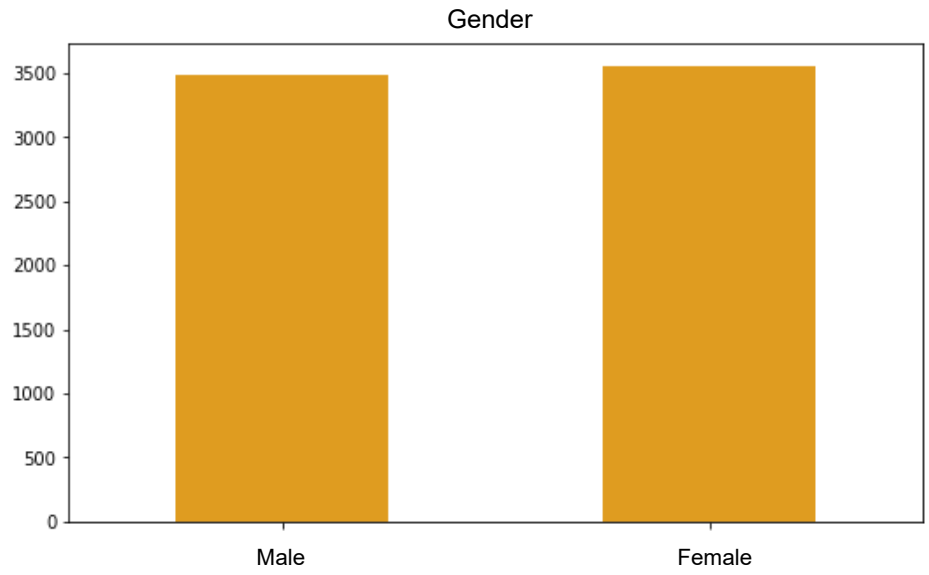


Figure 3.2: Gender.

Figure 3.3 depicts the three types of Internet services where it can be observed that the most used service is Fiber optic, and then DSL service.

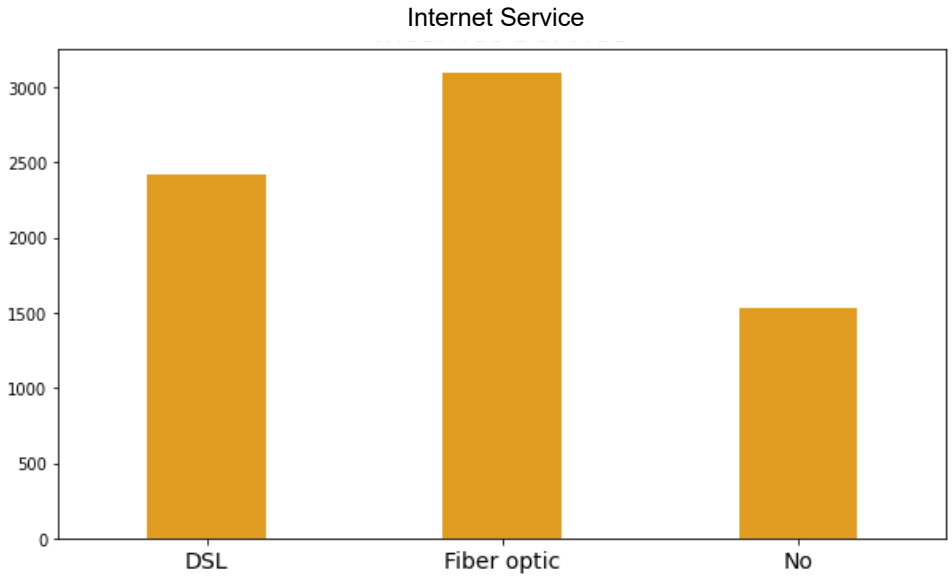


Figure 3.3: Internet Service.

Figure 3.4 the contract types are provided, where it can be observed that shorter contracts have a higher churn rate. Most customers must have a prepaid connection with the telecom operator.

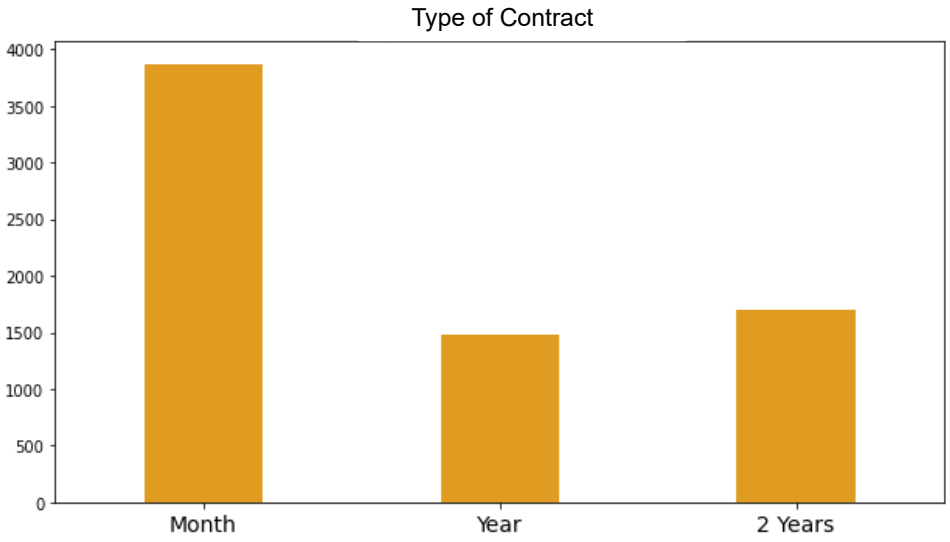


Figure 3.4: Types of Contract.

To analyse the maximum time a customer adheres to the service before churn, a time range function was created with intervals based on the number of months. As can be seen in Figure 3.5, most customers have contracts for up to one year.

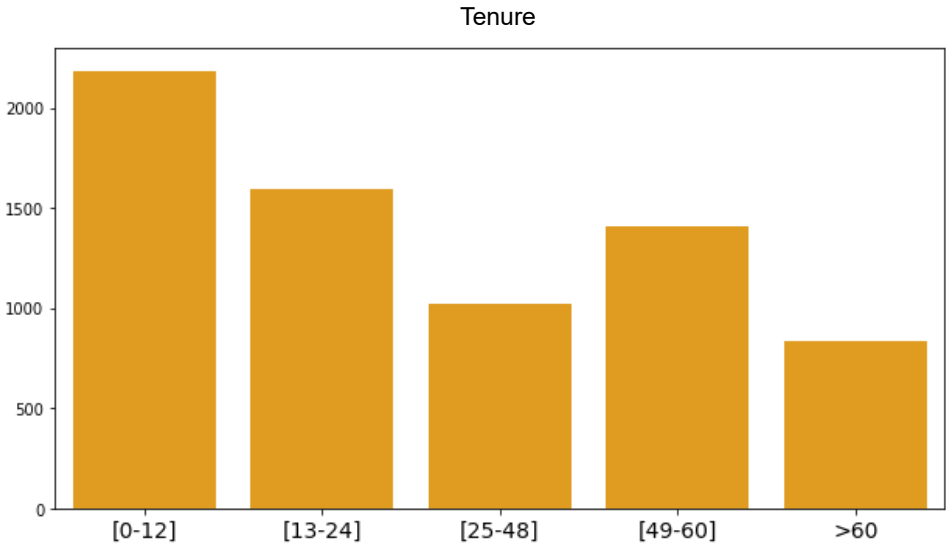


Figure 3.5: Customer Timeframe.

Figure 3.6 describes the form of payment used by clients, where it can be observed that customers prefer to pay their bills electronically, followed by check, wire transfer, and credit card payments.

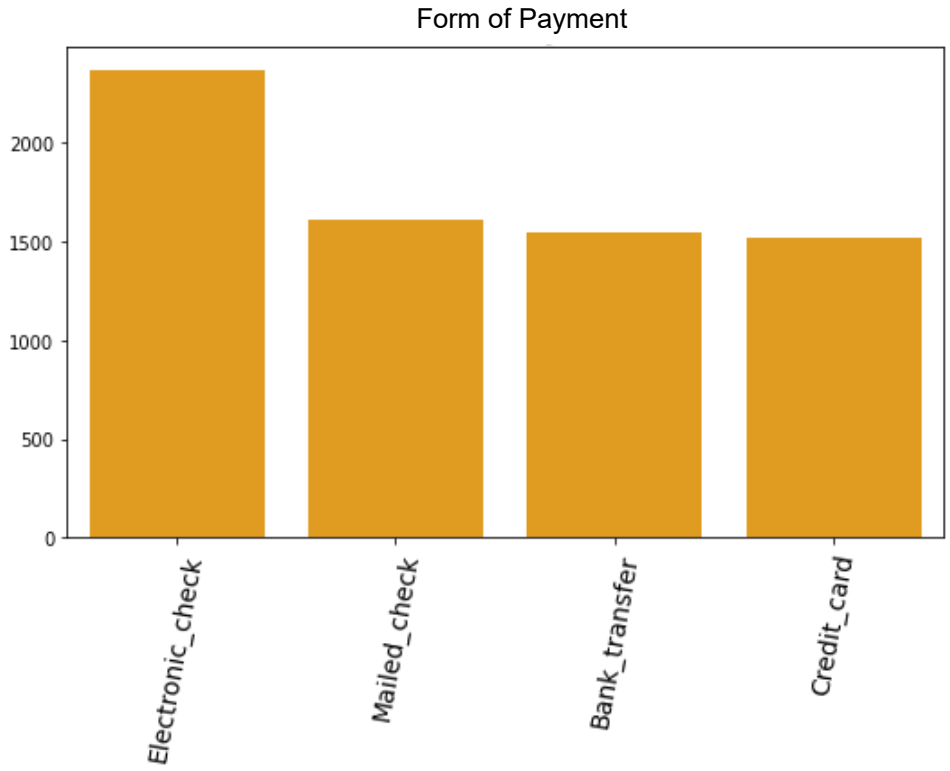


Figure 3.6: Payment methods.

Figure 3.7 depicts provides illustrates a distribution of positive (churn) and negative (non-churn) samples over the total charges provided in Figure (a). The same distribution is provided over monthly customer charges in Figure (b).

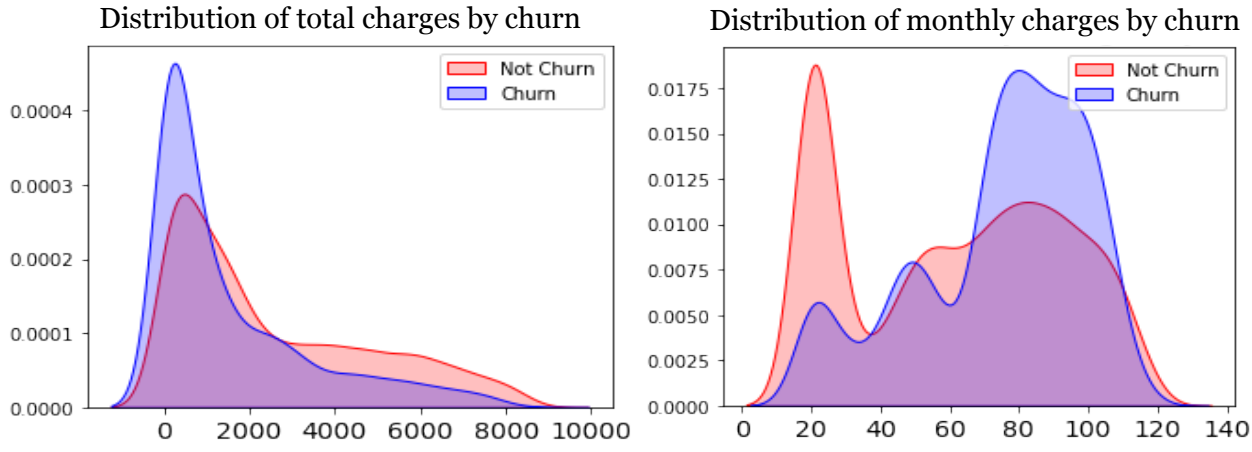


Figure 3.7: Distrubution of Total Charge and Monthly charge values.

Figure 3.8 there is the correlation matrix which is commonly used in building ML models to represent the correlation between different variables. In the figure below we have the heat map that was created to understand the linear relationship between the different variables in the churn rate dataset.

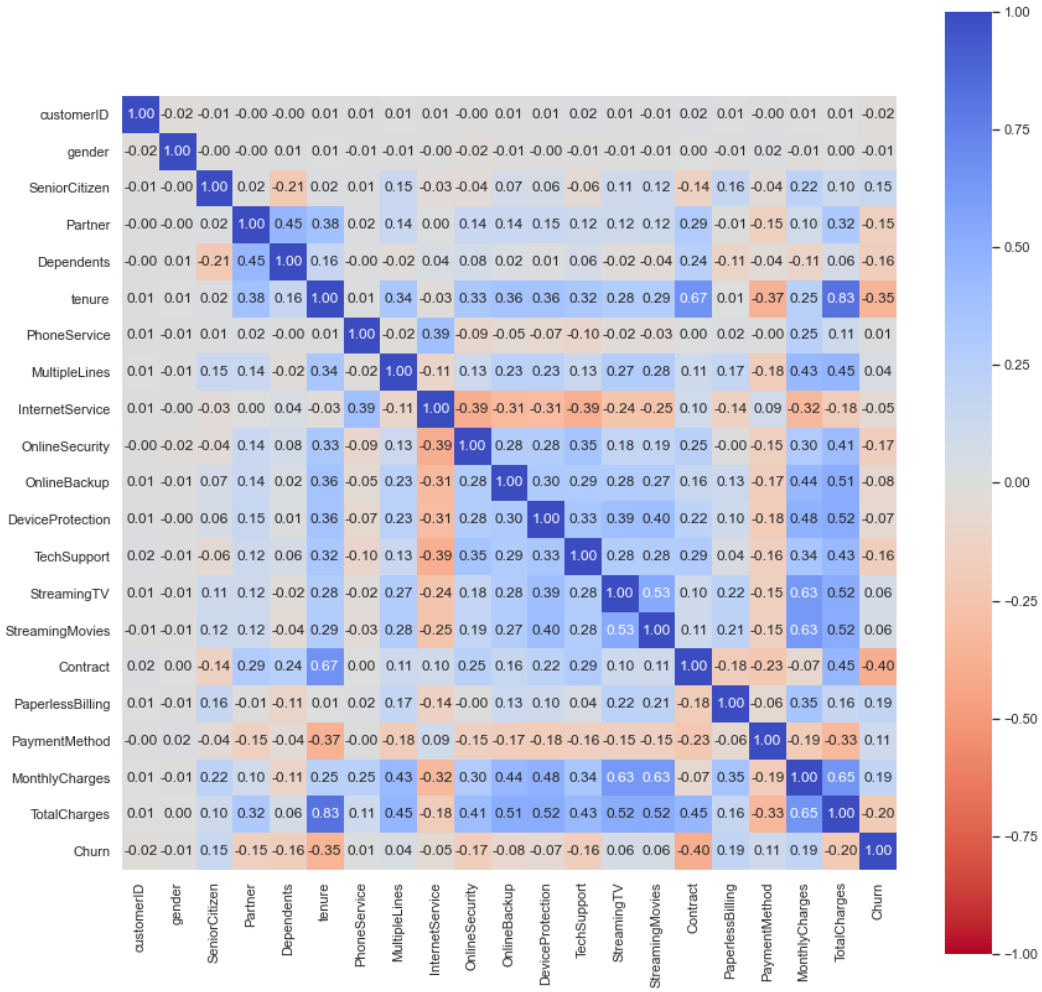


Figure 3.8: Correlation Heatmap for churn rate dataset.

Figure 3.9 we have the features that were classified in an importance ranking for churn prediction. This method was used to know which features are more relevant to the construction of the model.

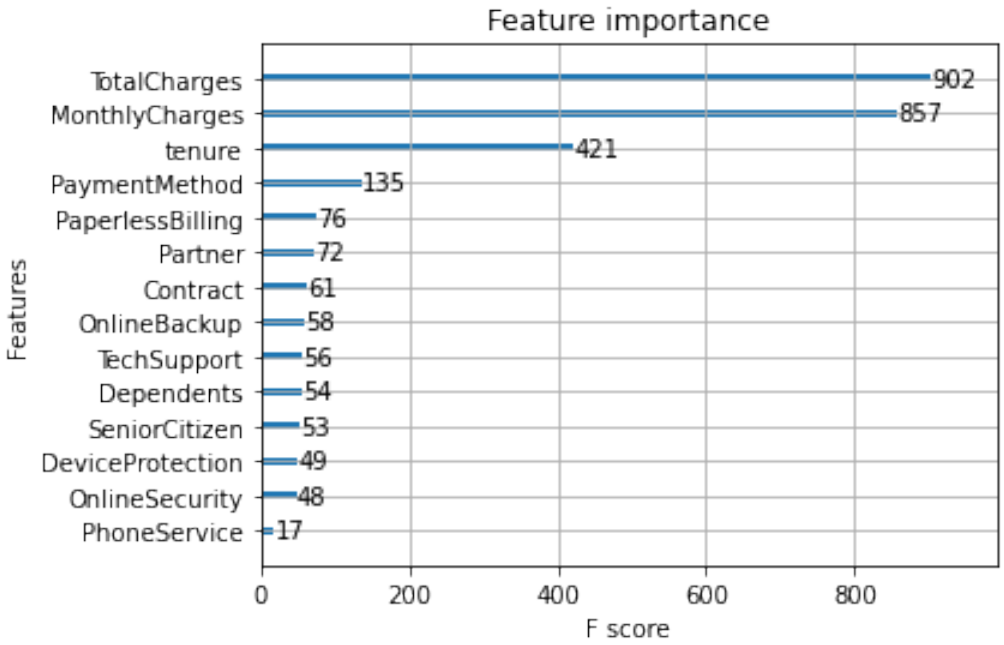


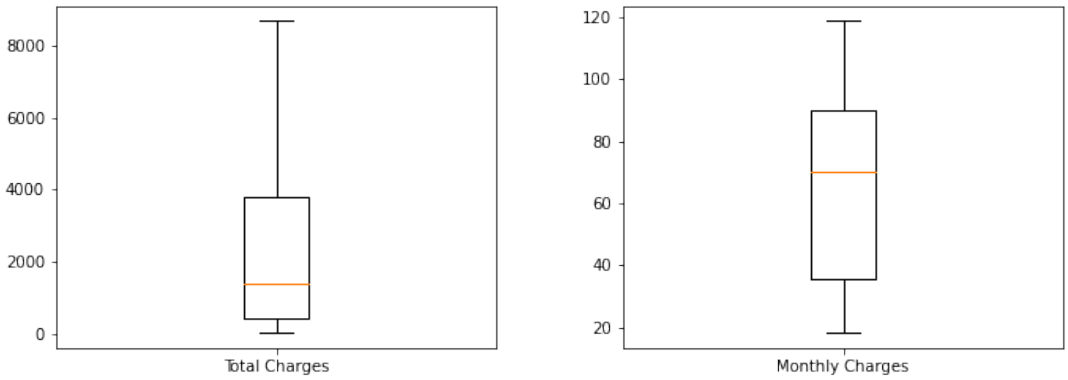
Figure 3.9: Feature Importance.

3.2.4 Feature Engineering

Feature Engineering is a component of pre-processing that consists of transforming raw data into useful data so that algorithms can interpret the data effectively. This process involves the selection, construction, transformation and extraction of features and all methods are applied according to the type of problem that is intended to be solved and also vary according to the different types of algorithms.

3.2.4.1 Outliers

The verification of outliers in data is an essential process, because it is through this method that it is possible to identify a data that is outside the global pattern of a distribution, i.e., a value that escapes normality. Identifying these values ensures that the performance of the models created are not affected. The interquartile range calculation was applied to the Churn rate dataset. The features analysed were the numerical ones, namely MonthlyCharges and TotalCharges. MothlyCharges represents the amount paid monthly by each customer. TotalCharges represents the total amount that has already been paid by a respective customer. In Figure 3.10 it is possible to observe that the applied calculation did not find outliers in the studied data set.



(a) (b)

Figure 3.10: Checking Outliers.

3.2.4.2 Missing Data

This is one of the mandatory steps in the process of data analysis, it directly influences the performance of ML models. Some algorithms are extremely sensitive to missing data, so their statistical power may be reduced and the results may present biased estimates. Therefore, it is important to pay attention to missing data in the design and execution of the data study. In Figure 3.11 we have the analysis of missing data that was performed on the Churn Prediction data set. As can be seen, the data set has no missing values, so it is not necessary to implement any imputation date technique.

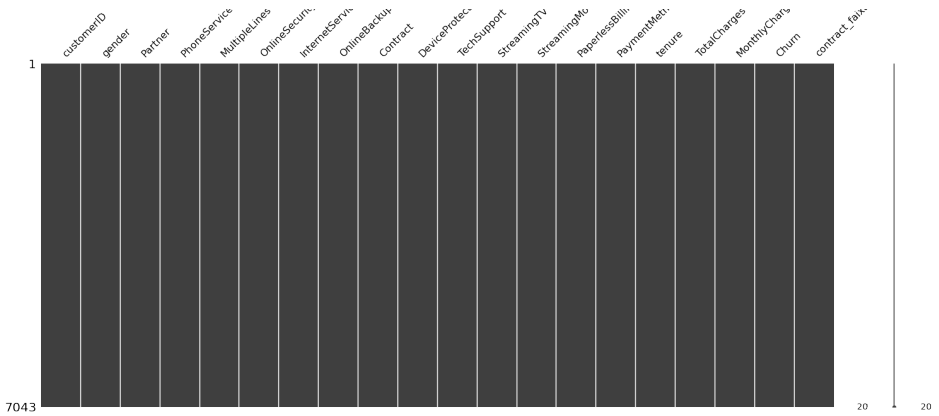


Figure 3.11: Checking for missing data in churn prediction.

3.2.4.3 Data Normalization

The normalization stage helps in the algorithm learning process speed, and consequently makes it present a better performance. In order to avoid that data were in different scales and that results were biased to variables with higher order of magnitude, normalization was applied to features Tenure, Monthly Charges and Total Charges. The method applied was the average normalisation that reduces the numerical value in a scale between -1.5

and 1.5.

3.2.4.4 Label Encoding

As it was verified, most of the data are of categorical type. The label encoding is responsible for transforming categories into numbers so that the algorithms can understand them. In Figure 3.12 we have the transformation that was performed on the data before building the ML models. This method was not applied for the features CustomerID, tenure, MonthlyCharges and TotalCharges.

customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...
712	0	0	1	0	1	0	1	0	0	...
2626	1	0	0	0	34	1	0	0	2	...
5572	1	0	0	0	2	1	0	0	2	...
3242	1	0	0	0	45	0	1	0	2	...
5254	0	0	0	0	2	1	0	1	0	...
6055	0	0	0	0	8	1	2	1	0	...
3567	1	0	0	1	22	1	2	1	0	...
25	0	0	0	0	10	0	1	0	2	...
6856	0	0	1	0	28	1	2	1	0	...
1965	1	0	0	1	62	1	0	0	2	...

Figure 3.12: Label Encoding applied to the dataset.

3.2.4.5 Imbalanced data

In Figure 3.13 we can observe that there is a severe imbalance where the minority data in the table is the one that informs when there was ”Churn”. And this pattern in the training data set can influence the algorithms to make mistakes, leading some to ignore the minority data class altogether. And this becomes a problem taking into consideration that it is the minority data class that matters in this context. To solve this problem it was used the $RandomUnderSample$ method that is able to reduce the number of examples of the majority class in the version of the training data set.

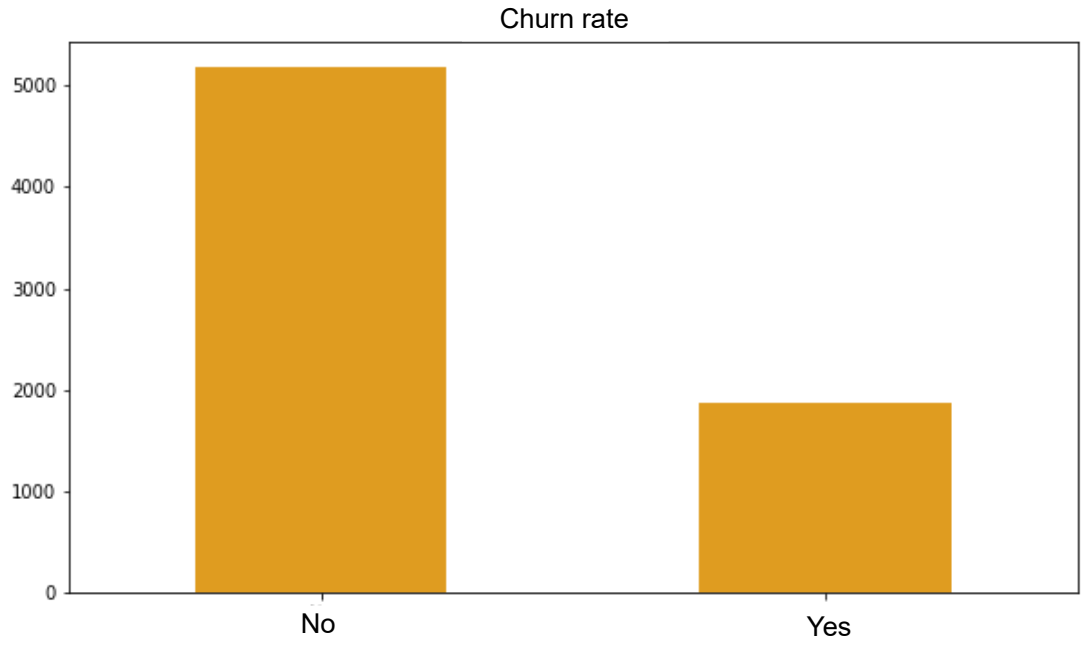


Figure 3.13: Churn Rate dataset.

3.3 Model Evaluation and Hyperparameter Optimization

Model evaluation is a crucial factor as a way to compare models, obtain performance metrics, and measure model stability. Models need to be accurate and capable of generalisation, i.e. Models are not fitted to a specific data set. This ensures that the model has captured most of the data patterns, and that underfitting or overfitting is not occurring. In modeling this problem, hyperparameters are used to help control the behavior of the training algorithms and make the ML models to achieve a more significant performance.

3.3.1 Cross-validation

Cross-validation is a technique capable of evaluating the generalisation ability of a model, from the dataset provided. This method is fundamental in predictive modelling problems. It seeks an estimate of how accurate a given model is in practice. Cross-validation maximises the number of data used for testing. To achieve this, the steps involved in cross-validation are: (1) reserve part of the sample; (2) use the remaining dataset and train the model; and (3) perform the test to the model using the reserved part of the dataset [Sab18]. Cross-validation is widely used to estimate the true prediction error of models and to tune model parameters. In this method the data are randomly divided into subsets called folds that are of equal (or nearly equal) sizes. Consecutively, K training and validation iterations are performed so that within each iteration, a different fold of data is kept for validation, while the remaining $k1$ folds are used in the learning process. Table 3.2 presents an example with K=5, where the darker part of the data is used for validation, while the lighter ones are used for training.

Table 3.2: 5-fold Cross-validation.

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Example 1
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Example 2
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Example 3
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Example 4
Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Example 5

3.3.2 Confusion Matrix

To evaluate the performance of the used models or the churn rate prediction in the testing dataset, different metrics have been used, namely recall, accuracy and f-measure [SJS06]. Through these metrics it is possible to analyze the ability of the predictive models to predict customer churn correctly [Ras15].
Figure 3.14 provides the confusion matrix which is an MxM matrix that is used to evaluate the performance of a classification model, where M is the number of target classes. Through the confusion matrix, it is possible to get a holistic view of the performance of classification models and the types of errors they may be making.

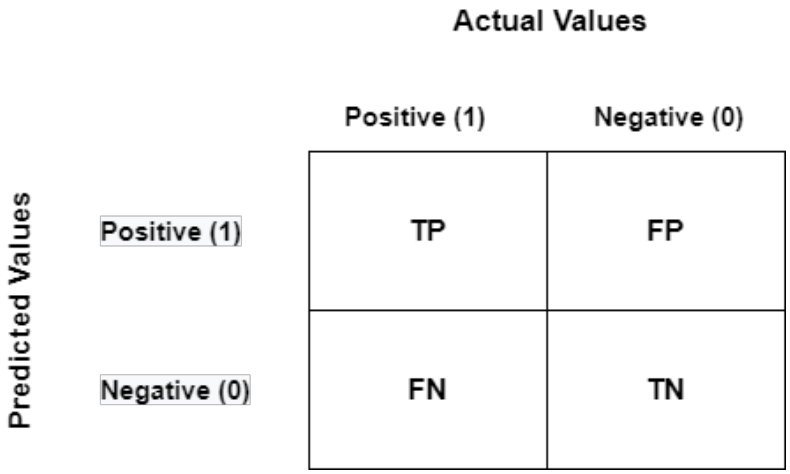


Figure 3.14: Confusion Matrix.

To evaluate the model's performance four criteria are used:

- **True positive (TP):** When the model is able to correctly predict customers who are in the termination category.
- **True negative (TN):** When the model is able to correctly predict customers who are in the non-dismissal category.
- **False positive (FP):** The number of customers who are non-churners but the predictive algorithm has labelled or identified them as churners.
- **False negative (FN):** The number of customers who are churners but the predictive model has labelled or identified them as non-churners.

Performance indicators

Recall: It is the proportion of true positive churners, the recall is calculated as following:

$$recall = \frac{TP}{TP + FN} \quad (3.1)$$

Precision: It is the ratio correct predicted churners, and it is calculated under the following:

$$precision = \frac{TP}{TP + FP} \quad (3.2)$$

Accuracy: It is ratio of number of all correct predictions, and it is calculated under the following:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.3)$$

F-measure: It is the harmonic average of precision and recall, and it is calculated under the following:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3.4)$$

3.3.3 Receiver Operating Characteristic Curve

Another way to measure performance in classification problems is the AUC ROC curve which is built on the confusion matrix. ROC is a curve that calculates the probability, the relationship between TPR and FPR, its benefits and costs. While AUC makes the representation of the degree or measure of separability, that is, it tells how much the model is able to distinguish between the existing classes. Literature explains that the higher the AUC, the better the model can predict churn and no churn customers for example, because AUC provides an aggregate measure of performance across all possible classification boundaries. As we can see, the ROC curve analysis is a powerful tool to select optimal models and discard models that do not correspond with the results we seek. In Figure 3.15 it can be seen that the excellent curve is at coordinates (0,1) in the ROC space. In this case the closer the green line is to 1 in the Y axis, the better the model will be in discriminating between positive and negative. The point (0,1) is also called ”perfect classification” [FUW06].

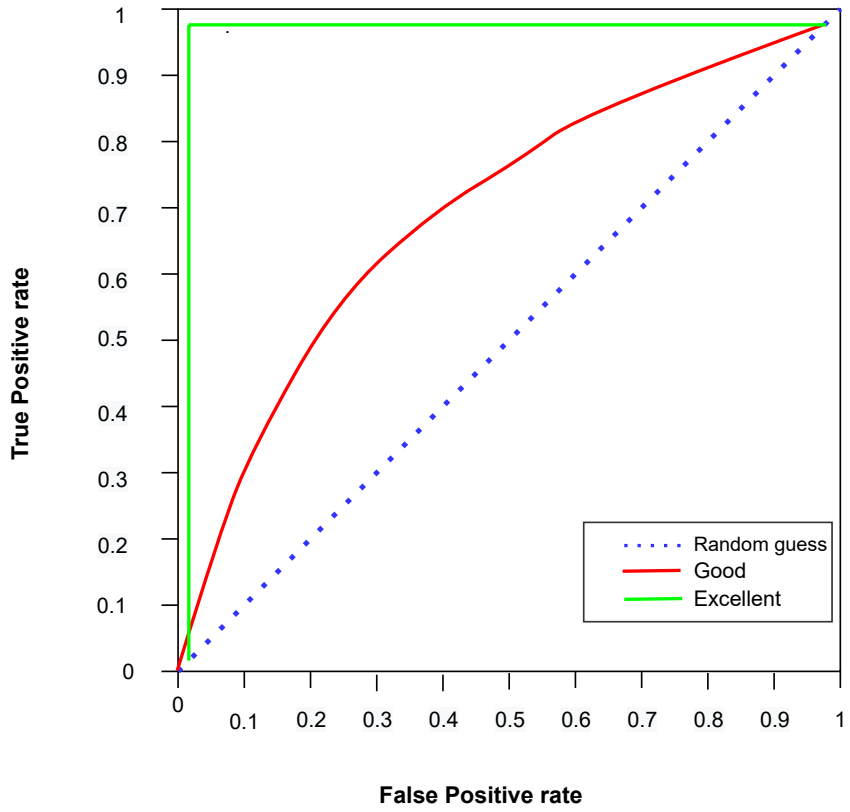


Figure 3.15: ROC Curves.

3.3.4 Mean Squared Error

The MSE is a way to analyze the performance of the models that obtained better predictive ability. It is called MSE because it is able to find the mean square difference between the estimated values and what is estimated. The formula for the MSE is shown below:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (3.5)$$

Where $\hat{Y}_i$ is the predicted value of the samples and Y_i is the vector of observed values of the variable being predicted. In this metric, the closer the score is to 0, the better the prediction is. The MSE calculation was performed in all models that were developed in this research.

3.3.5 Hyperparameter Optimization

In ML the term hyperparameter is used to differentiate the parameters of the standard model. The adjustment of hyperparameters is one of the factors that help to improve the model and make it possible to obtain a more satisfactory performance of the model. Hyperparameter optimization gives us the possibility of building more flexible models with the possibility of setting different values, and performing the choice of values that test bet-

ter accuracy. The authors Bergstra, Bardenet, Bengio, and Kégl [BBBK11], presented two main algorithms that perform searches to optimize hyperparameters, namely grid search and random grid search. In the research of this thesis, grid search was used. The GridSearchCV method is a function that comes in the $model\_selection$ package of Scikit learn. The method was an essential factor in optimizing the hyperparameters. The systematic method is able to go through all the parameters that are listed. The main goal was to try to improve the model in all its possible ways, many attempts and tests were made.

3.4 Conclusion

This chapter presented one of the most important processes in the construction of ML models. Through EDA, graphics were generated to help understand how categorical and numerical values are distributed in the dataset and how variables are correlated with each other and what patterns can be extracted from these data. In feature engineering it was possible to analyze the data in a deeper way by checking for outliers, missing data, unbalanced values and perform the transformation of categorical values into binary values so that the algorithms follow to obtain a good performance. Another very important point of this chapter is the methodology used to analyze the results of the models that were built. Each method has the function of measuring the performance of each model and ensuring that it is performing the best possible result of accuracy.

Chapter 4

Building the Churn Rate Models

4.1 Introduction

This chapter covers how the implementation process of the Churn prediction models was carried out using all the algorithms that were defined in Section 3. A diagram is presented that demonstrates how the step-by-step of the proposed models was built. The section 4.3 defines the most relevant core technologies and libraries that have been used. In section 4.4 the pseudocode of the algorithm that is proposed to build the churn prediction is presented and the same is applied for all 10 models. A brief definition was made about the hyperparameters that were defined in the turing with the GridsearchCV method.

1. 4.2 - **Model Proposal** - Schematic of the model building process and datasets used.

2. 4.3 - **Techonologies and Librares** - A brief summary of all the most relevant technologies that were used in this study.

3. 4.4 - **Implementation Details** - Details of model construction and the hyper parameters used.

4. 4.5 - **Conclusion** - Summary of all the points that have been covered in this chapter.

4.2 Model Proposal

In Figure 4.1, we have the explanatory diagram of the steps that were necessary for the modelling of Churn Prediction, we can observe that in the image the training and test data were divided 80% and 20% respectively, soon after that the algorithms are trained with the training data using cross validation, this is done to avoid overfitting the model and to increase its generalization. The hyperparameter optimization was performed with the goal of bringing better results to the classifiers. After training the model receives the test data that is unknown to it, where it will be tested and analyzed how good the model is through the module evaluation that was mentioned in section 3.3.

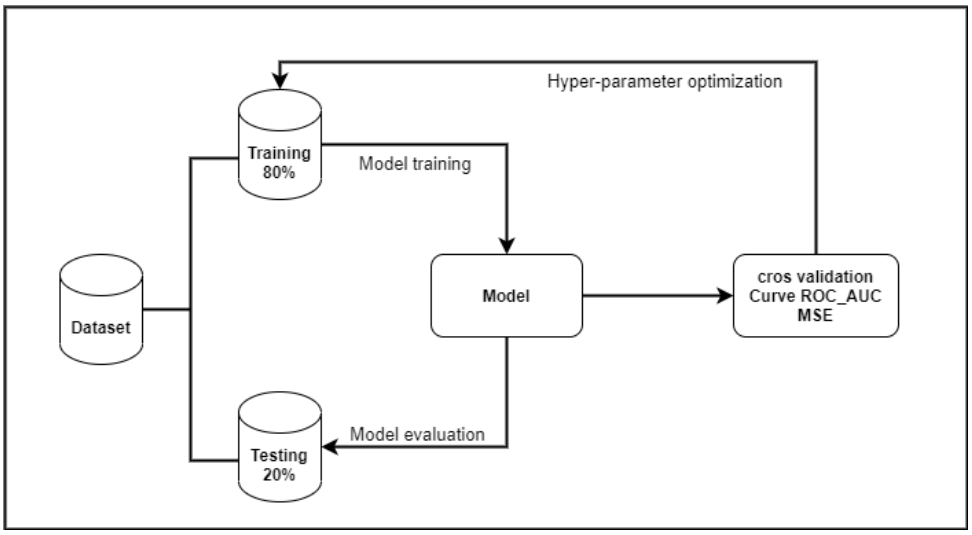


Figure 4.1: Diagram of the proposed model.

4.2.1 Dataset Used

In this research we used three datasets with different types of CCP features, the two datasets that are presented below are from the Kaggle dataset. On each CCP dataset 10 different algorithms were implemented and the results of this implementation are presented in Chapter 5. Table 4.1 has dataset 2 where the features and their types are presented. This dataset is composed of 21 features and mostly of numeric values, quite different from dataset 1 where most features are categorical type.

Table 4.1: Dataset 2 Features and their types.

Features	**Types**
State	categorial
Account Length	numeric
Area Code	numeric
Phone	numeric
Int'l Plan	categorial
VMail Plan	categorial
VMail Message	numeric
Day Mins	numeric
Day Calls	numeric
Day Charge	numeric
Eve Mins	numeric
Eve Calls	numeric
Night Mins	numeric
Night Calls	numeric
Intl Mins	numeric
Intl Calls	numeric
CustServ Calls	numeric
Churn	boolean

Table 4.2 has dataset 3 that is composed of 14 features, most of the values in this dataset are also numeric. It should be noted that in all the datasets the feature engineering process was implemented, where all the necessary treatments on the data were done, from its

normalization to the exclusion of features that were not necessary in the modeling.

Table 4.2: Dataset 3 Features and their types.

Features	**Types**
RowNumber	numeric
CustomerId	numeric
Surname	object
CreditScore	numeric
Geography	object
Gender	categorial
Age	numeric
Tenure	numeric
Balance	numeric
NumOfService	numeric
HasCrCard	numeric
IsActiveMember	numeric
EstimatedSalary	numeric
Churn	numeric

4.3 Techonologies and Librares

The python programming language was used to develop the ML models. The choice of this language was due to the fact that it is a simple and intuitive language, besides providing a series of tools that assist in the development of ML models. For the development of predictive models, a virtual environment was created with jupyter notebook, which is an open source web application. The jupyter notebook can be used for all kinds of data science tasks, it proved to be very versatile and flexible throughout the process. To build the models some libraries were used, in this list are the most relevant ones. The most used was the scikit-learn library for having a range of useful tools for building predictive models. Below we have the definition of each one.

- **XGboost** - Implements ML algorithms under the Gradient Boosting framework, it is a library that is designed to be highly efficient, flexible and portable. It is considered an algorithm that solves data science problems in a fast and accurate way [xd].

- **Catboost** - is an open source algorithm for gradient boosting on decision trees. The algorithm is widely used in the construction of predictive models for achieving a good performance in their predictions.

- **LightGBM** - has a gradient boosting framework that uses tree-based learning algorithms. It has a number of advantages, among them are: lower memory usage, parallel learning support and is able to handle a large amount of data [Cor].

- **Scikit-learn** - is an open source ML library for Python programming language. It is capable of implementing classification algorithms, SVM, gradient boosting, among

others [OGM]. This was the most used library in this work, both to import algorithms and to report some specific metrics.

- **Statistics** - This module provides specific functions for performing statistical math calculations with numerical value data [Fou].

- **Imblearn.under_sampling** - This library was used to handle the data with unbalanced values, it is responsible for doing a random subsampling on the majority class [GL].

4.4 Implementation Details

To describe in more detail how the construction of the ML models was performed, we have pseudocode 1 below which presents the procedures used in the implementation. An exhaustive search for the best hyperparameters using GridSearchCV was performed. This was one of the crucial strategies to drive the algorithms to achieve considerable performance. Section 5 exposes the influence that this optimization has on the classifiers.

Algorithm 1 Proposed Algorithm for Churn Prediction

1: Classifier labels for test instances
2: The train data-set consisting of input features x and output label y;
3: Predict Labels (churn or non-churn);
4: **Procedure**
5: Identification of most suitable data using data analysis techniques;
6: Cleaning and filtering (handling null and missing values);
7: Balancing data using random under-sampling;
8: Tuning the use of hyper-parameters using exhaustive search;
9: Application predictive models using Adaboost, Naive Bayes, Random Forest ...
10: Evaluation of the results using confusion matrix, curve AUC and MSE.

In ML for each algorithm there is a vast possibility of exploring its hyperparameters, which give us numerous opportunities to perform optimization in the models and application of different techniques. In order to convey knowledge about the hyperparameters that were used in the churn rate modeling, below is a list with a brief description of each of them.

- **max_features** - The number of features to consider when searching for the best split of a node.

- **ccp_alpha**- Cost complexity pruning parameter, can be used to control tree size.

- **max_depth**- The maximum depth of the tree. Setting this parameter prevents overfitting from occurring.

- **criterion**- Responsible for determining how the impurity of a division will be measured.

- **max_leaf_nodes**- Sets the best nodes as relative reduction of impurities.

- **random_state** - Selects a random combination of training and test data.
- **n_estimators** - Number of trees you want to build in the model.
- **learning_rate** - Considered a key hyperparameter, it is responsible for adjusting the step size of each iteration of the algorithm while moving towards a minimum value of a loss function.
- **num_leaves** - This parameter is responsible for controlling the complexity of the mod link, it defines the maximum number of leaves each weak learner has. If this parameter is set to a high value it may suffer overfitting.
- **penalty** - Defines which type of regularization the algorithm will undergo if it is L1 or L2.
- **solve** - Defines the algorithm that will be used for the model optimization, *newton-cg*, *lbfgs*, *liblinear*, *sag*, *saga*. Each one has a different purpose that varies depending on the amount of data.
- **leaf_size** - Determines the size of the algorithm and influences its performance.
- **n_neighbors** - Defines the number of neighbors that will be used in the algorithm.
- **p** - Power parameter for the minkowski metric.
- **weights** - Determines the weight function that will be used in the prediction which are *uniform* or *distance*.
- **metric** - In this parameter the distance metric to be used for the tree is defined, namely *minkowski* or *chebyshev*.
- **eval_metric** - Determines the metric that is used to evaluate the model at each iteration. Importantly, this parameter does not guide the optimization.
- **C** - Regularization parameter of the algorithm to avoid misclassification of each training data set.
- **gamma** - Responsible for defining how much curvature we want on a decision boundary.
- **kernel** - Determines what type of kernel will be used by the algorithm.
- **var_smothing** - Determines the float value that will be used to calculate the largest variations for each feature and add it to the stability calculation variation [Pyt].
- **n_jobs**- Determines whether parallel jobs should be allowed. If assigned the value 1, no parallel jobs are used, if set to 1, all CPUs are used.

4.5 Conclusion

This chapter covered the entire process of building the machine learning models, exposing the model scheme, talking about how the data were divided, and about the optimization that was performed using the hyperparameters that exist in the algorithms, precisely to help us build effective and simple models. The additional datasets present in the tables (4.1, 4.2) were used as a way to obtain more concise results about the models that were built and also as a way to generate a certain degree of comparison in how the models behave with different types of data. Technologies and libraries that are constantly used by data scientists seeking to solve supervised learning type problems were used. After an exhaustive search for the optimization of the models, searches were performed on each algorithm and then hyperparameters that aim to significantly improve their performance were chosen.

Chapter 5

Results and Discussion

5.1 Introduction

In this chapter, we provide the results and discussions about the performance of the models with and without the optimization of the hyperparameters on the three datasets that were detailed in this research. The final results are obtained using the, such as accuracy, precision, recall and f1-score of each model. It also contains details about which values were assigned to the hyrperparameters that each algorithm received. Results of the curve of AUC, MSE and discussion about the performance that the models obtained.

1. 5.2 - **Model Result** - Performance results of the 10 models that were implemented with and without hyperparameters.
2. 5.3 - **AUC Curve and MSE Results** - The graphs of the AUC that the models were able to achieve and the MSE values of each prediction are displayed.
3. 5.5 - **Discussion** - The results are compared with each other and with the datasets that were used.
4. 5.6 - **Conclusion** - Summary of all the points that have been covered in this chapter.

5.2 Model Result

After the three datasets that were referenced in this research have gone through all the procedures mentioned in the 3 and 4 section, the datasets are fitted to each model with its default setting and tested against the test data. Then the CV Gridsearch method is applied with cross-validation that varies from 5 to 10 times these values are fitted with the purpose of finding the best hyperparameters for the models we are building. This search was done in all datasets despite the computational cost. The datasets 3.1 and 4.2 are relatively large and this requires a higher data processing power. In order to understand the difference between before and after using the hyperparameters for model optimization, we also report which hyperparameters were used in each algorithm cited in 2.8.

5.2.1 Random Forest

First the RF algorithm tested only with its default configuration obtained the results that can be found in the table below 5.1.
Although the hyperparameters have not yet been defined the model with the RF algorithm showed promising performance with the test data. After running the grid search with CV=10, the following parameters were found:

Table 5.1: RF metrics before GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	79	89	87
Precision (%)	82	97	88
Recall (%)	91	89	96
F1-Score(%)	86	93	92

- max_depth = 70
- random_state=44
- max_features=auto
- criterion = entropy
- ccp_alpha=0.001

Table 5.2: RF metrics after GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	92	95	77
Precision (%)	95	98	94
Recall (%)	91	90	76
F1-Score(%)	86	94	84

5.2.2 Decision Tree

With its normal standardization the DT algorithm obtained an excellent performance in relation to the test data of Data_01, however when adjusted with the hyperparameters the results were different and the score dropped. When the hyperparameters were applied on the dataset Data_02 e Data_03 we were able to increase the score ranging from 2 to 3 points. We can analyze the details in tables 5.3 and 5.4.

Table 5.3: DT metrics before GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	87	82	70
Precision (%)	94	97	91
Recall (%)	80	82	69
F1-Score(%)	86	89	78

After running the grid search with CV=10, the following parameters were found:

- max_depth = 15
- max_leaf_nodes=89
- criterion = entropy

- ccp_alpha=0.001

Table 5.4: DT metrics after GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	75	84	73
Precision (%)	77	97	93
Recall (%)	72	84	72
F1-Score(%)	74	90	81

5.2.3 AdaBoost

AdaBoost is an algorithm widely used in classification problems precisely because of its strategy of combining weak learning with a single strong classifier. In building this model the AB was implemented with its default parameters and then added the DT algorithm along with parameter adjustment, we can see the results in the tables below 5.5 and 5.6.

Table 5.5: AB metrics before DT and GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	78	81	86
Precision (%)	83	95	89
Recall (%)	89	82	94
F1-Score(%)	86	88	92

After running the grid search with CV=10, the following parameters were found:

- random_state=42
- max_leaf_nodes=89

Table 5.6: AB metrics after DT and GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	87	84	72
Precision (%)	91	96	92
Recall (%)	83	84	71
F1-Score(%)	87	89	80

5.2.4 kNN

KNN was one of the algorithms that its results were directly driven in all three datasets by the adjustment of the hyperparameters. We can observe that Data_02 obtained a score in which the variation was minimal if compared to the result presented in 5.20

Table 5.7: KNN metrics before GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	69	87	76
Precision (%)	75	97	81
Recall (%)	88	87	96
F1-Score(%)	81	92	91

After running the grid search with CV=10, the following parameters were found:

- leaf_size= 20
- n_neighbors= 10
- P= 1
- weights= distance
- metric= minkowski

Table 5.8: KNN metrics after GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	88	83	72
Precision (%)	91	96	92
Recall (%)	83	84	71
F1-Score(%)	87	89	80

5.2.5 XGBoost

We can consider that this is one of the most versatile algorithms and its library implements the gradient boosting decision tree algorithm. It presented promising results even before the hyperparameters adjustments, achieving high scores.

Table 5.9: XGB metrics before GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	86	87	85
Precision (%)	92	97	89
Recall (%)	79	88	94
F1-Score(%)	85	92	91

After running the grid search with CV=5, the following parameters were found:

- learning_rate= 0.3
- max_depth= 15
- n_estimators = 100

Table 5.10: XGB metrics after GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	89	92	86
Precision (%)	94	98	89
Recall (%)	84	93	94
F1-Score(%)	89	96	91

5.2.6 Naive Bayes

NB is a fast algorithm and has the ability to handle large volumes of data. This algorithm showed little improvement in its results even after the hyperparameters adjustment.

Table 5.11: NB metrics before GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	72	84	79
Precision (%)	88	96	81
Recall (%)	73	85	86
F1-Score(%)	79	90	88

After running the grid search with CV=10, the following parameters were found:

- var_smoothing = 1e-08

Table 5.12: NB metrics after GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	76	83	72
Precision (%)	90	93	90
Recall (%)	76	86	72
F1-Score(%)	82	90	80

5.2.7 CatBoost

Catboost is an algorithm in which the library is based on gradient boosting and has a wide variety of hyperparameters that can be adjusted, and it can easily handle categorical values. The model presented excellent results in Data_01 e Data_02. In Data_03, the score tended to decrease after going through GridsearchCV.

Table 5.13: CatBoost metrics before GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	79	89	86
Precision (%)	83	98	89
Recall (%)	89	90	95
F1-Score(%)	86	93	92

After running the grid search with CV=5, the following parameters were found:

- learning_rate = 0.5
- max_depth = 8
- n_estimators = 100
- random_state=44

Table 5.14: CatBoost metrics after GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	95	90	77
Precision (%)	96	97	94
Recall (%)	97	91	76
F1-Score(%)	97	94	84

5.2.8 Logistic Regression

LR is an algorithm used to solve binary classification problems. Despite being a widely used model, it did not perform well on Data_01 e Data_02. In Data_03, the model was able to recover after adjusting the hyperparameters.

Table 5.15: LR metrics before GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	75	74	79
Precision (%)	77	95	80
Recall (%)	72	74	97
F1-Score(%)	75	83	88

After running the grid search with CV=5, the following parameters were found:

- penalty= l1
- solver= liblinear
- C=1

Table 5.16: LR metrics after GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	78	76	81
Precision (%)	83	94	83
Recall (%)	89	77	96
F1-Score(%)	86	85	89

5.2.9 SVM

SVM is an algorithm that does not require a large computational power, making the tests to run faster. The model created was able to achieve great results on Data_01 and Data_02, but it was less than perfect when applied to Data_03. We can observe its performance in the tables below.

Table 5.17: SVM metrics before GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	65	85	45
Precision (%)	64	97	86
Recall (%)	72	85	38
F1-Score(%)	68	91	52

After running the grid search with CV=10, the following parameters were found:

- kernel= rbf
- gamma= 1
- C=1

Table 5.18: SVM metrics after GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	94	85	77
Precision (%)	91	97	80
Recall (%)	99	84	95
F1-Score(%)	95	90	87

5.2.10 LGBM

LGBM is also one of the gradient boosting and decision tree based algorithms. It is a great classifier to solve churn rate problem. LGBM proved to be a powerful algorithm for its speed and high performance. After the adjustments in hyperparameters the model was able to achieve a higher score.

Table 5.19: LGBM metrics before GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	78	82	85
Precision (%)	83	97	86
Recall (%)	88	82	97
F1-Score(%)	86	89	91

After running the grid search with CV=10, the following parameters were found:

- learning_rate= 0.3

- max_depth= 1
- n_estimators= 100
- num_leaves= 50

Table 5.20: LGBM metrics after GridsearchCV

Churn Data	Data_01	Data_02	Data_03
Accuracy (%)	91	95	87
Precision (%)	94	95	89
Recall (%)	86	99	95
F1-Score(%)	90	97	92

5.2.11 Confusion Matrix

Confusion Matrix is one of the strategies used in this research to further analyze the performance of each classifier and find out if the model predicts well the class we want (churn). Understanding the distribution of TP, TN, FP and FN rates is essential to employ improvements on the models and understand how the values are being classified. After the 10 models were tested on the three datasets presented, tables of the best performing models were generated. In Figure 5.6 we have the tables referring to the results of Data_01. As a point to be observed, the Error Rate (EER) was chosen, where the number of all incorrect predictions divided by the total number of the dataset is calculated, and thus we find the model with the lowest error rate. The model that showed the lowest error rate in this dataset was the LGBM Classifier and Random Forest, surprised with a rate of only 0.09. LGBM proved to be a fast algorithm that has a great compatibility with large datasets, the classifier was able to achieve the lowest error rate in all datasets. The second model with the lowest EER is XGBoost which obtained a rate of 0.10. The worst EER result was with the KNN model which obtained 0.13. In Figure 5.2 we have as the best performance the LGBM model with EER of 0.05 and then XGBoost with only 0.06. These are excellent results and indicate that the models are erring infrequently. The third model with the lowest EER was Random Forest and then CatBoost which were between the values 0.08 and 0.09 respectively. AdaBoost was the model with the worst rate, scoring 0.16. In Figure 5.3 we have Data_03 that in comparison with the other results was the dataset that obtained higher scores in the EER rate. In this dataset the model with the lowest rate was LGBM that scored 0.12 then Random forest, CatBoost and AdaBoost that scored EER of 0.13. The worst result was from the KNN model that scored 0.21, we can state that this would not be a model chosen to classify churn prediction.

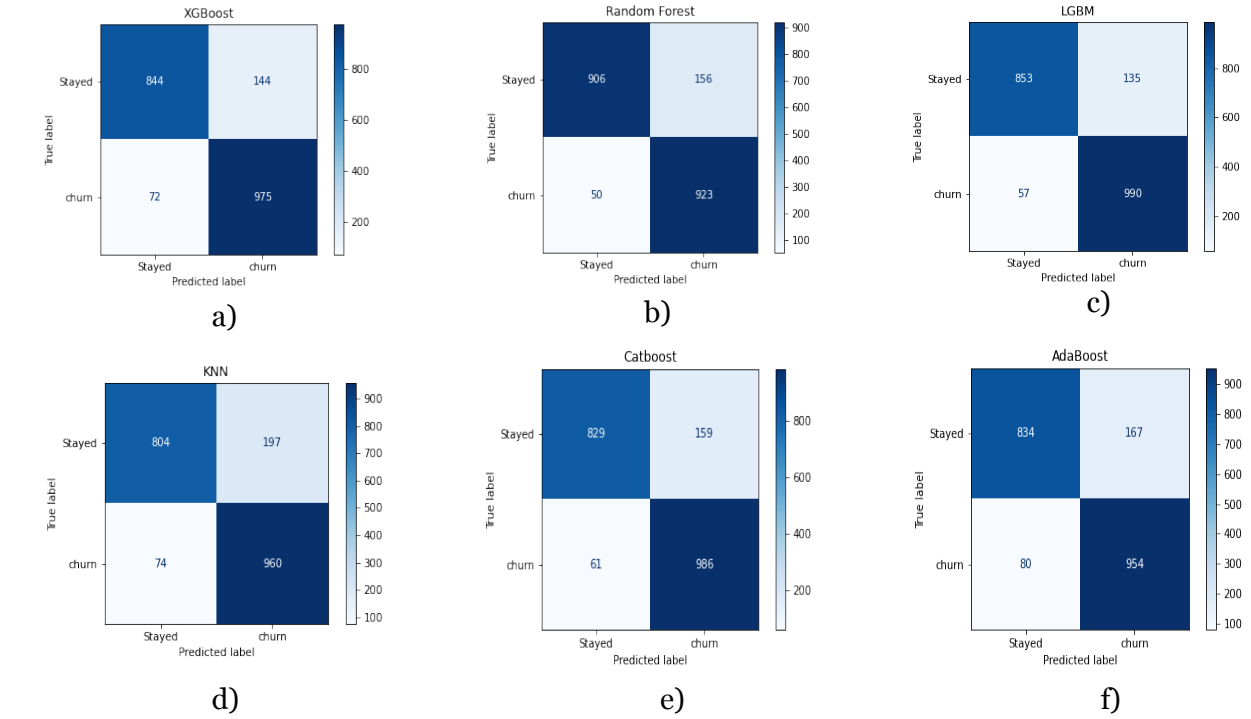


Figure 5.1: Confusion Matrix Data_01.

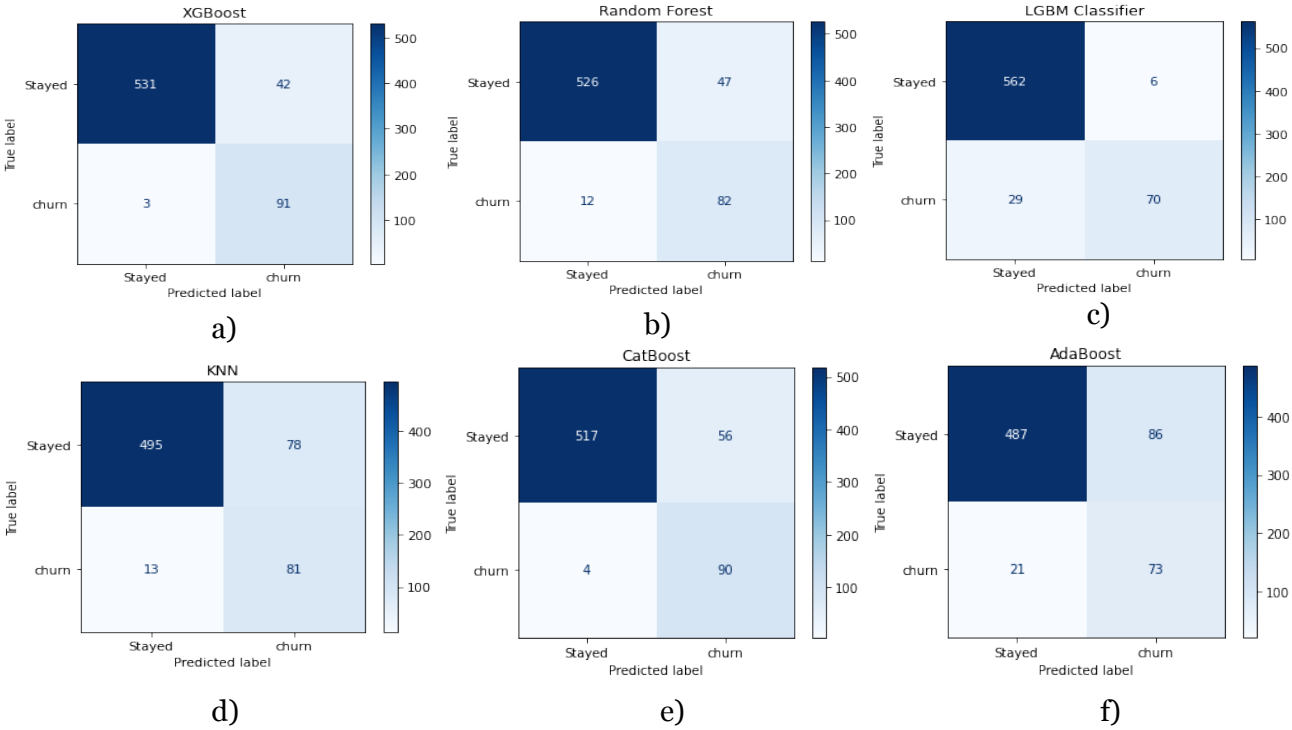


Figure 5.2: Confusion Matrix Data_02.

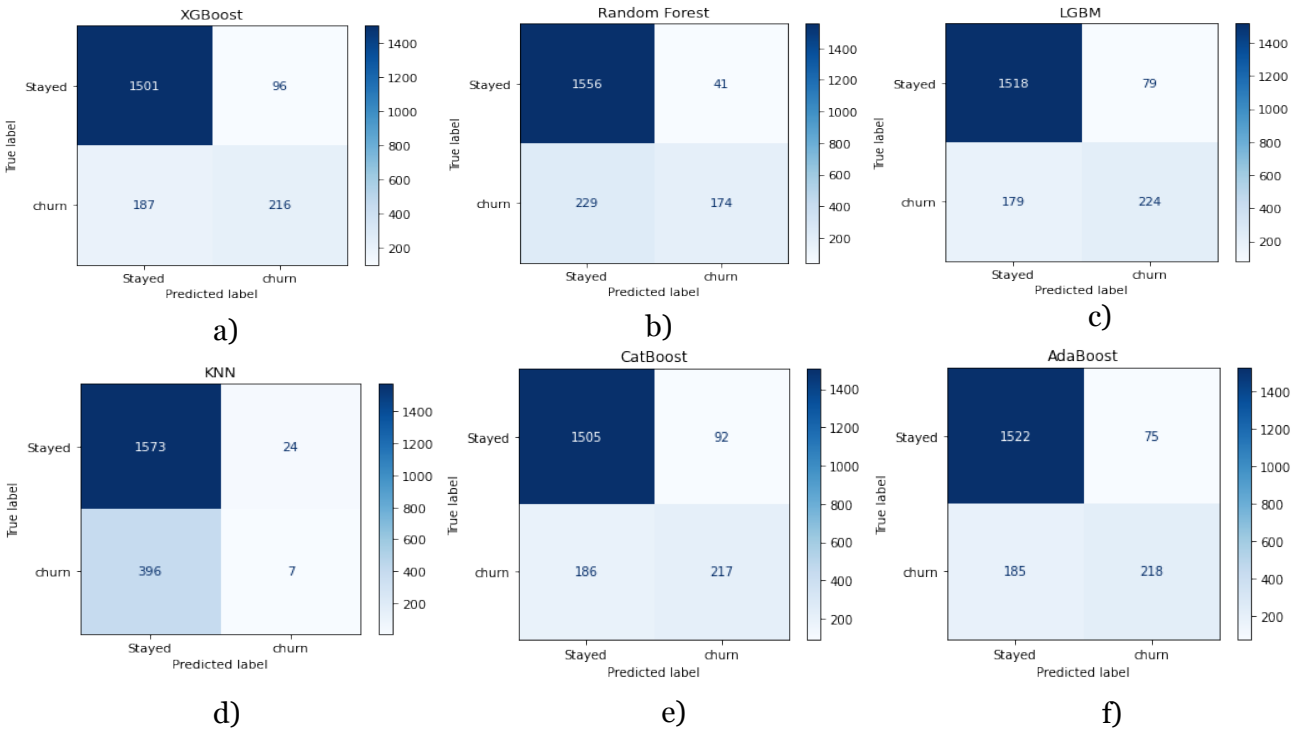


Figure 5.3: Confusion Matrix Data_03.

5.3 AUC Curve Analysis

As commented earlier in Section 3.3.3 the Curve ROC is a commonly used way to visualise the performance of a binary algorithm. From AUC curve it is possible to analyse how well the classifier separated the two classes. Making a brief observation of the AUC curve of the three datasets below, we can notice a pattern that was also noticed in the study that was developed based on the related works. The classifiers that use the ensemble method obtained a much more significant performance when compared to the other algorithms that were used to predict Churn. This technique consists in the combination of weak learner with strong learner, which aims at obtaining a model that is more consistent and with less vulnerable to noise. The Data_01 considered the main dataset of this research obtained the highest score with AUC of 98% using the RF algorithm, the developed model outperformed the results that were presented by [LMCS21] where in his research using the same dataset the RF classifier obtained AUC of 82%. Another pattern found in the results is that the CatBoost, LGBM Classifier and XGBoost classifiers which are based on gradient boosting (GB) obtained a very good AUC score with 92%, 91% and 93% respectively, also outperforming the results found in [LMCS21], where the XGBoost and CatBoost models found AUC values of 84% and 82% respectively. The worst AUC curve result in this dataset was with the SVM classifier that scored AUC score of 81%, and despite this also outperformed the result of [LMCS21] where the model scored 79%. In Data_02 the RF algorithm also scored the highest AUC score with 92%, the gradient boosting algorithms scored similar scores of 91% in each model. The worst result in this dataset was with Logistic Regression that scored AUC of 82%. In Data_03 the results were frustrating, because

some models could not reach a good performance having a disastrous result, not being able to achieve the necessary generalization. An example of this was the KNN model that despite having obtained a good performance in datasets 01 and 02 was not enought with a poor score of 56% and the SVM that scored AUC of 60%. The best result in Data_03 was the score obtained by CatBoost with AUC of 88%, the other models managed to achieve scores around 86%.

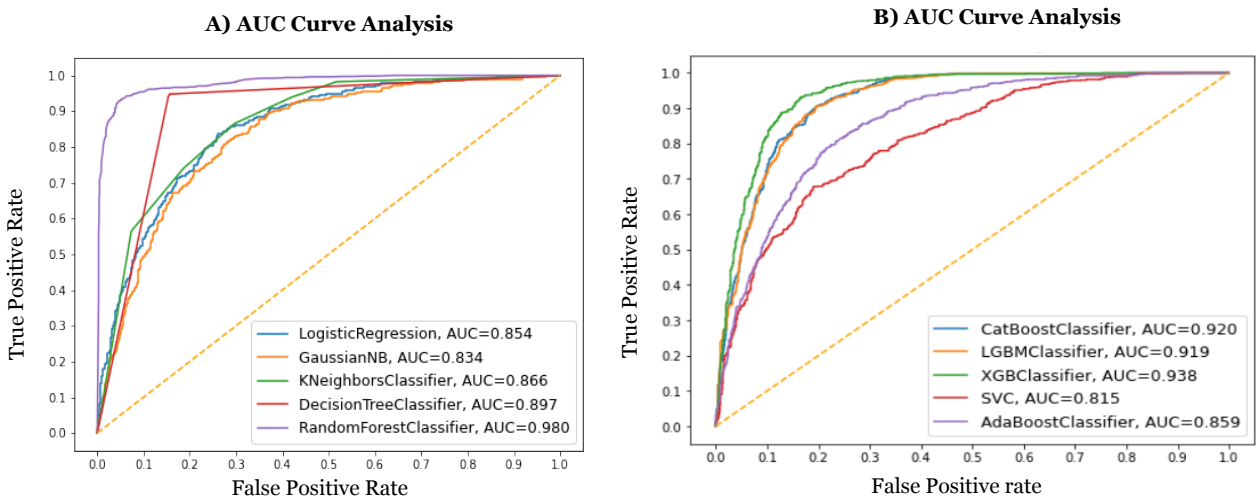


Figure 5.4: AUC Curve Analysis Data_01.

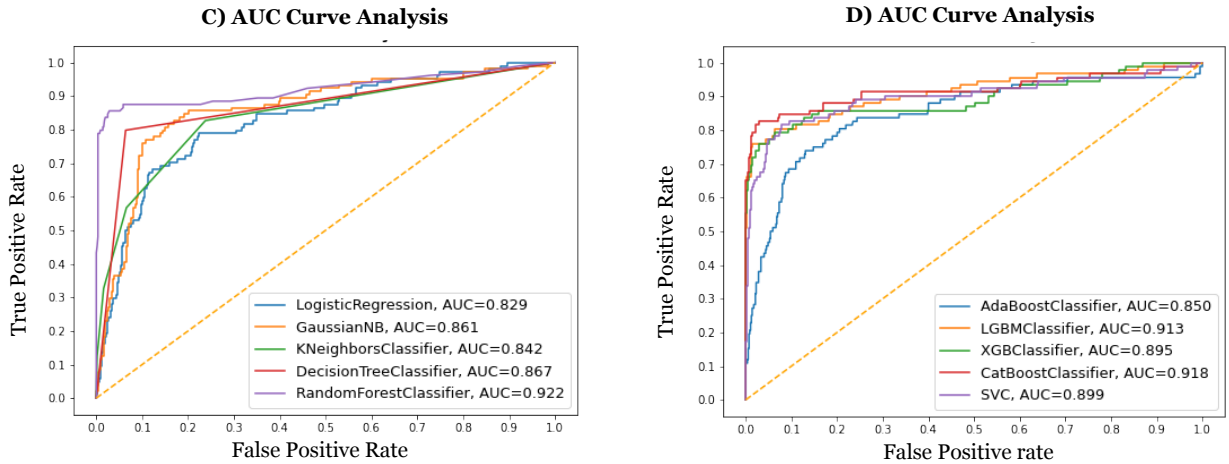


Figure 5.5: AUC Curve Analysis Data_02.

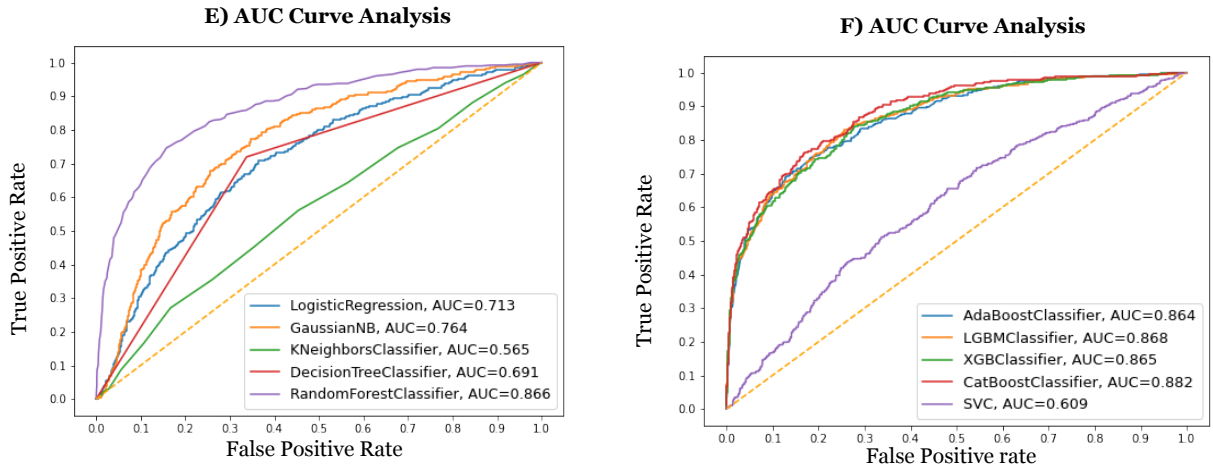


Figure 5.6: AUC Curve Analysis Data_03.

5.4 MSE Analysis

This is a standard metric used in ML to find how much the model does not agree with the actual data. In the Table 5.4 we have the values that the models were able to obtain. The smaller the value, better the adjustment to the model. In the Data_01 we have a tie in the scores that use the ensemble method. Random Forest, XGBoost and LGBM Classifier were able to obtain the standard deviation of 0.11, followed by KNN with MSE of 0.13. In Data_02 the best performance was with the algorithm LGBM Classifier that once again is proving to be an algorithm with great potential for classification, the MSE value of this model was 0.05. CatBoost, XGBoost and Random Forest were able to obtain values of 0.09,0.08,0.09 respectively. The worst result in this data set was with the Logistic Regression that achieved a score of 0.24. In Data_03 we can see that also the best results were also from gradient boosting and Random Forest models. In both Data_01 and Data_03 the Naive Bayes model however, did not perform so well in this section.

Table 5.21: MSE of different models on datasets.

MSE	**Data_01**	**Data_02**	**Data_03**
Random Forest	0.11	0.09	0.14
Decision Tree	0.23	0.16	0.23
AdaBoost	0.22	0.16	0.13
KNN	0.13	0,13	0.21
XGBoost	0.11	0.08	0.14
CatBoost	0.21	0.09	0.13
Logistic Regression	0.22	0.24	0.19
SVM	0.06	0.15	0.23
LGBM Classifier	0.11	0.05	0.14
Naive Bayes	0.28	0.17	0.28

5.5 Discussion

As it was possible to observe, the results of the models were explored in different ways of evaluation. The investigation of these results using the techniques that have been mentioned in this chapter is of utmost importance to build a model that can really solve the problem and create opportunities for the sector in which it is desired to apply, in addition to being able to better understand its operation and how its performance can be boosted. Based on the studies applied in related works to solve the churn prediction problem it can be observed that the solutions obtained by the authors are strictly related to the nature of the problem data. And this is also something that we can analyze in this research, when applying the models that were built in different datasets it was possible to observe that always the models that use bagging and boosting techniques managed to achieve a high perfomance, this can also be visualized in the research performed by [URM+19], [AJA19], where models like Random Forest and XGBoost dominate the scores. These techniques are promising. In the accuracy issue, the models that obtained better results in all datasets were Random Forest, AdaBoost, Catboost, KNN, XGBoost, SVM and LGBM Classifier, and this was only possible after the hyperparameters were adjusted. After using this technique it was possible to understand that to build a good model it is not necessary to apply any complexity on it, but to understand the problem and use the tools that ML itself provides us, performing a simple modeling on the problem. In [BS19] a comparison between KNN and Logistic Regression was performed. The results of this study expressed that the Logistic Regression scored better in terms of accuracy and AUC when compared to the KNN model. Therefore, it can be concluded that for the type of data presented in this study, Logistic Regression is more effective in predicting customer churn compared to KNN, which was proven in the results obtained in this research. In this context of discussion the questions that were exposed in 1.2 are answered.

1. What is the most appropriate machine learning model to use to predict future customer churn?

   R: As confirmed in this study, the best performing classifiers are those that implement ensemble methods such as bagging and gradient boosting algorithms, namely Random Forest, LGBM Classifier, Catboost and XGBoost. These algorithms can achieve good performance with fast speed and low computational cost. Catboost achieved an accuracy of 95%, Random Forest got 92% and LGBM 91%.

2. Which features can be considered the most important to build a predictive customer churn model?

   R: The most relevant features are those with values that contain information about the amounts the customer pays monthly and the total amount already paid to the operator, in addition the patterns can also be seen in the type of service the customer uses.

3. What are the possible solutions to deal with unbalanced classes?

R: In binary classification problems it is common to have unbalanced classes and there are different techniques to solve this type of problem. In this study the technique known as Random Under-sampling was used to balance the data set of the majority class. Performing this process is extremely important to obtain a better performance of the ML model.

4. What impacts can the predictive model have on the telecommunications industry?

   R: The impact on the telecom industry is quick and direct. It can be said that churn is seen as a thermometer for the company's health, and so understanding what generates churn is important to: Identify difficulties for business growth; Build a more agile business model; Maximize the value that the customer has within the brand; Increase customer engagement, among other points. The predictive models proposed in this research is able to solve one of the problems inherent to the telecommunications industry and make it increase its profits by up to 85%. Furthermore, the company that seeks to implement this type of technology has a competitive advantage over other industries in the sector and can easily increase its profits. Marketing strategies can be proposed according to consumer behavior and also reward those with a low probability of churn, as well as positively influence the brand name in the market.

5.6 Conclusion

This chapter has shown all the results that the models were able to obtain, the process of which hyperparameters were used in each algorithm, and how the evaluation was performed on each model and what metrics were chosen to perform the comparisons between them. It was possible to analyze the models that obtained better performance and get relevant information of which can be considered the best ones to solve the problem. The results were compared with the results that of the works mentioned in 2 and the questions that were raised in the 1 chapter were answered.

Chapter 6

Conclusion

One of the main characteristics of the telecommunications sector is its standardization and public telecommunications policies that allow customers to switch operators very easily, resulting in a competitive market. Therefore, churn prediction, or the task of recognizing customers who are likely to stop using the service, is an essential factor for the profitability aspects of the overall telecommunications sector. The modeling for this type of problem is not standardized, since there is no approach that is the key solution to solve the telecom service providers turnover problem worldwide. Data set analysis and machine learning techniques are used to predict churn based on customer history, giving telecom companies the opportunity to take preventive measures such as customized marketing strategies and thus increase their revenues.
This study analysed the performance of ten different types of algorithms on three different datasets and with this we obtained valuable conclusions on how modelling can be adjusted to solve this type of problem. To test and build the model, the sample data was split into 80% for training and 20% for testing. We chose to perform the cross-validation varying from 5 to 10 folds, for hyperparameter optimization, and for calculation of the error rate of each model and MSE. The techniques of feature engineering, effective feature transformation and selection approach were applied so that the models could use the data in an effective way, and the problem of unbalanced data was solved by subsampling or using tree algorithms that are not affected by this problem. It was proved that for this classification problem the models that use the ensemble method can obtain a satisfactory performance if compared with models of the decision tree or naive bayes type, for example. In Data_01 and Data_02, the other models generated good results with only small differences in the performance metrics. In Data_03 as the nature of the data was a little different some models, for example SVM could not achieve the supposed result.

6.1 Contributions and Achievements

Based on the solution developed in this work and in order to materialize the research project that has a partnership with the research and development company TimweLab Tech, the models that obtained the best performance were deployed. Below we have the figure 6.1 that represents how the models were put into production. In the figure we have the Cloud, the Python server where the ML models are implemented, which connects to the backend that encompasses everything that serves the applications, both the mobile applications of the Telco Vista project and the Backoffice platform. Our Web API's, databases and Unomi are found there. Unomi is a REST server that manages client profiles and events that are related to each profile. It is a server that can easily

interact with external systems, promoting the sharing of profiles and reuse in different applications. All services and implemented code can be found in the GitHub repository `https://github.com/AnaHauachen` open to all the community that wishes to be inspired by this research.

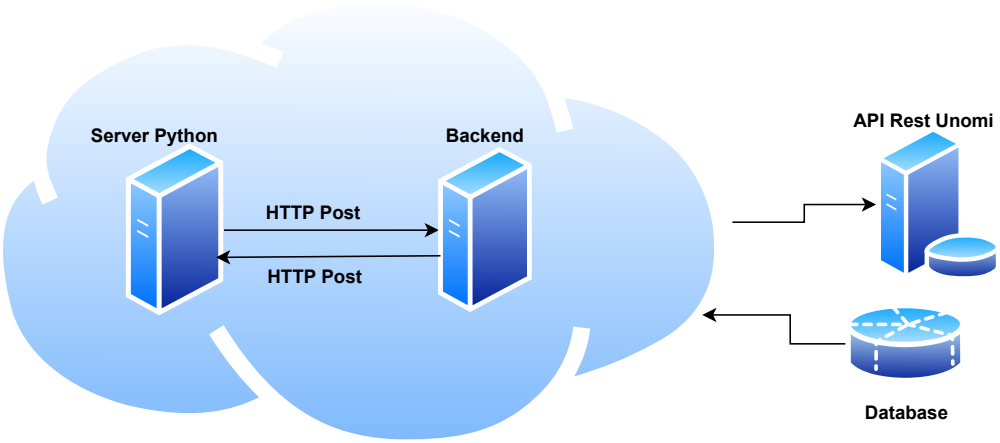


Figure 6.1: Architecture of the model deployment.

Bibliography

[AH01] Selim Aksoy and Robert M Haralick. Feature normalization and likelihood-based similarity measures for image retrieval. *Proceedings of the Pattern recognition letters*, 22(5):563–582, 2001. 8

[AJA19] Abdelrahim Kasem Ahmad, Assef Jafar, and Kadan Aljoumaa. Customer churn prediction in telecom using machine learning in big data platform. *Proceedings of the Journal of Big Data*, 6(1):1–24, 2019. viii, 19, 57

[Ale21] Aleksey Bilogur. Misingno documentation [online]. 2021. Available from: `https://pypi.org/project/missingno/#description` [cited 21 Junho 2021]. 25

[AMR17] Mohd Khalid Awang, Mokhairi Makhtar, and Mohd Nordin Abdul Rahman. Improving accuracy and performance of customer churn prediction using feature reduction algorithms. *Proceedings of the Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 9(2-3):127–130, 2017. 20

[AST11] Hossein Abbasimehr, Mostafa Setak, and MJ Tarokh. A neuro-fuzzy classifier for customer churn prediction. *International Journal of Computer Applications*, 19(8):35–41, 2011. vii, 1

[BB12] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012. 11

[BBBK11] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. *Proceedings of the Advances in neural information processing systems.*, 24, 2011. 37

[BGM+20a] P. Bhuse, A. Gandhi, P. Meswani, R. Muni, and N. Katre. Machine learning based telecom-customer churn prediction. In *Proceedings of the 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, pages 1297–1301, 2020. vii, 1

[BGM+20b] Pushkar Bhuse, Aayushi Gandhi, Parth Meswani, Riya Muni, and Neha Katre. Machine learning based telecom-customer churn prediction. In *Proceedings of the 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, pages 1297–1301. IEEE, 2020. 16

[BN19] Niranjanamurthy M Bhawna Nigam, Himanshu Dugar. Effectual predicting telecom customer churn using deep neural network. *Engineering and Advanced Technology (IJEAT)*, 8(2):103–112, 2019. 19

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 13

[BS19] A. Bhatnagar and S. Srivastava. A robust model for churn prediction using supervised machine learning. In *Proceedings of the 2019 IEEE 9th International Conference on Advanced Computing (IACC)*, pages 45–49, 2019. 20, 57

[BTB16] Ionuţ Brânduşoiu, Gavril Toderean, and Horia Beleiu. Methods for churn prediction in the pre-paid mobile telecommunications industry. In *Proceedings of the 2016 International conference on communications (COMM)*, pages 97–100. IEEE, 2016. 20

[Bur19] Andriy Burkov. *The hundred-page machine learning book*, volume 1. Andriy Burkov Canada, 2019. 6

[CDM15] Marc Claesen and Bart De Moor. Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*, 2015. 11

[Cor] Microsoft Corporation [online]. Available from: `https://lightgbm.readthedocs.io/en/latest/`. 41

[DCCDB18] Arno De Caigny, Kristof Coussement, and Koen W De Bock. A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *Proceedings of the European Journal of Operational Research*, 269(2):760–772, 2018. 20

[EOUD17] UF Eze, CJ Onwuegbuchulam, CH Ugwuishiwu, and S Diala. Application of data mining in telecommunication industry. *International Journal of Physical Sciences*, 12(6):74–88, 2017. 11

[FH19] Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated machine learning*, pages 3–33. Springer, Cham, 2019. 11

[Fou] Python Software Foundation [online]. Available from: `https://docs.python.org/3/library/statistics.html`. 42

[FUW06] Jerome Fan, Suneel Upadhye, and Andrew Worster. Understanding receiver operating characteristic (ROC) curves. *Canadian Journal of Emergency Medicine*, 8(1):19–20, 2006. 35

[GFH09] Salvador García, Alberto Fernández, and Francisco Herrera. Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems. *Proceedings of the Applied Soft Computing*, 9(4):1304–1314, 2009. 8

[GL] D. Oliveira C. Aridas. G. Lemaitre, F. Nogueira [online]. Available from: `http://glemaitre.github.io/imbalanced-learn/generated/imblearn.under_sampling.RandomUnderSampler.html`. 42

[Hil12] Constantinos S Hilas. Data mining approaches to fraud detection in telecommunications. In *2nd PanHellenic Conference on Electronics and Telecommunications-PACET'12*, 2012. 12

[IT21] IT. Matplotlib documentation [online]. 2021. Available from: `https://matplotlib.org/stable/contents.html#` [cited 20 Junho 2021]. 25

[JKS20] Hemlata Jain, Ajay Khunteta, and Sumit Srivastava. Churn prediction in telecommunication using logistic regression and logit boost. *Procedia Computer Science*, 167:101–112, 2020. 20

[K+97] Philip Kotler et al. Marketing management: Analysis, planning, implementation and control. 1997. vii, 1

[Kar98] Andrew H Karp. Using logistic regression to predict customer retention. In *Proceedings of the Eleventh Northeast SAS Users Group Conference. http://www. lexjansen. com/nesug/nesug98/solu/p095. pdf*, 1998. 15

[KJ+13] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013. 10

[KNN16] Ledisi G Kabari, Domaka N Nanwin, and Edikan Uduak Nquoh. Telecommunications Subscription Fraud Detection Using Naïve Bayesian Network. *International Journal of Computer Science and Mathematical Theory*, 2(2), 2016. 12

[Lau20] Julius Lauw. An Information-Theoretic Perspective on Overfitting and Underfitting. In *AI 2020: Advances in Artificial Intelligence: 33rd Australasian Joint Conference, AI 2020, Canberra, ACT, Australia, November 29-30, 2020, Proceedings*, volume 12576, page 347. Springer Nature, 2020. 8

[LM98] Huan Liu and Hiroshi Motoda. Feature transformation and subset selection. *IEEE Intell Syst Their Appl*, 13(2):26–28, 1998. 8

[LMCS21] Praveen Lalwani, Manas Kumar Mishra, Jasroop Singh Chadha, and Pratyush Sethi. Customer churn prediction system: a machine learning approach. *Computing*, pages 1–24, 2021. 20, 54

[LP03] Junxiang Lu and O Park. Modeling customer lifetime value using survival analysis—an application in the telecommunications industry. *Data Mining Techniques*, pages 120–128, 2003. 1

[MA16] Rayan Masoud and Tarig Mohamed Ahmed. Using data mining in telecommunication industry: Customer's churn prediction model. *Journal of Theoretical and Applied Information Technology*, 91(2):322, 2016. 12

[MC13] Walid Moudani and Fadi Chakik. Fraud detection in mobile telecommunication. *Lecture Notes on SOftware Engineering*, 1(1):75, 2013. 12

[MTMM13] Golshan Mohammadi, Reza Tavakkoli-Moghaddam, and Mehrdad Mohammadi. Hierarchical neural regression models for customer churn prediction. *Journal of Engineering*, 2013, 2013. vii, 1

[Num21] Numpy. Numpy documentation [online]. 2021. Available from: `https://numpy.org/doc/stable/user/whatisnumpy.html` [cited 26 Agosto 2021]. 25

[OGM] Jérémie du Boisberranger Olivier Grisel, Guillaume Lemaitre and Chiara Marmo [online]. Available from: `https://scikit-learn.org/stable/`. 42

[Pan21] Pandas. Pandas [online]. 2021. Available from: `https://pandas.pydata.org/` [cited 26 Agosto 2021]. 25

[Pen09] Parag C Pendharkar. Genetic algorithm based neural network approaches for predicting churn in cellular wireless network services. *Expert Systems with Applications*, 36(3):6714–6720, 2009. vii, 2

[PGV+17] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. CatBoost: unbiased boosting with categorical features. *arXiv preprint arXiv:1706.09516*, 2017. 18

[PXBJ13] Li Peng, Yu Xiaoyang, Sun Boyu, and Huang Jiuling. Telecom customer churn prediction based on imbalanced data re-sampling method. In *Proceedings of 2013 2nd International Conference on Measurement, Information and Control*, volume 01, pages 229–233, 2013. 9

[Pyd] Seaborn Pydata [online]. Available from: `https://seaborn.pydata.org/`. 25

[Pyt] Holy Python [online]. Available from: `https://holypython.com/nbc/naive-bayes-classifier-optimization-parameters/`. 43

[QRQ+13] Saad Ahmed Qureshi, Ammar Saleem Rehman, Ali Mustafa Qamar, Aatif Kamal, and Ahsan Rehman. Telecommunication subscribers' churn prediction model using machine learning. In *Eighth international conference on digital information management (ICDIM 2013)*, pages 131–136. IEEE, 2013. 12

[Ras15] Sebastian Raschka. *Python machine learning*. Packt publishing ltd, 2015. 5, 34

[Reg21] Reginaldo J. Santos. Xgboost documentation [online]. 2021. Available from: `https://xgboost.readthedocs.io/en/latest/` [cited 20 Junho 2021]. 16

[RF99] JoséLuis Rodriguez-Fernández. Ockham's razor. *Endeavour*, 23(3):121–125, 1999. 9

[Sab18] Sahar F Sabbeh. Machine-learning techniques for customer retention: A comparative study. *Proceedings of the International Journal of advanced computer Science and applications*, 9(2), 2018. 14, 15, 16, 19, 33

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 7

[Sch13] Robert E Schapire. Explaining adaboost. In *Empirical inference*, pages 37–52. Springer, 2013. 17

[SJS06] Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*, pages 1015–1021. Springer, 2006. 34

[sph21] sphinx. Lgbmclassifier [online]. 2021. Available from: `https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html` [cited 26 Agosto 2021]. 18

[SZ06] Jiang Su and Harry Zhang. A fast decision tree learning algorithm. In *AAAI*, volume 6, pages 500–505, 2006. 15

[Tho92] Christopher James Thornton. *Techniques in computational learning: An introduction*. Chapman & Hall Computing, 1992. 8

[UI16] V Umayaparvathi and K Iyakutti. A survey on customer churn prediction in telecom industry: Datasets, methods and metrics. *Proceedings of the International Research Journal of Engineering and Technology (IRJET)*, 3(04), 2016. viii

[URM+19] Irfan Ullah, Basit Raza, Ahmad Kamran Malik, Muhammad Imran, Saif Ul Islam, and Sung Won Kim. A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector. *Proceedings of the IEEE Access*, 7:60134–60149, 2019. 19, 20, 57

[VDA16] Wil Van Der Aalst. Data science in action. In *Process mining*, pages 3–23. Springer, 2016. 8, 25

[VDSC15] Thanasis Vafeiadis, Konstantinos I Diamantaras, George Sarigiannidis, and K Ch Chatzisavvas. A comparison of machine learning techniques for customer churn prediction. *Proceedings of the Simulation Modelling Practice and Theory*, 55:1–9, 2015. viii

[WC02] Chih-Ping Wei and I-Tang Chiu. Turning telecommunications call details to churn prediction: a data mining approach. *Expert systems with applications*, 23(2):103–112, 2002. 1

[Wei05] Gary M Weiss. Data mining in telecommunications. In *Data Mining and Knowledge Discovery Handbook*, pages 1189–1201. Springer, 2005. 11

[WM94] Janusz Wnek and Ryszard S Michalski. Hypothesis-driven constructive induction in AQ17-HCI: A method and experiments. *Machine Learning*, 14(2):139–168, 1994. 8

[xd] xgboost developers [online]. Available from: `https://xgboost.readthedocs.io/en/latest/`. 41

[YRS] Mohamed Khaled Yaseen, Mafas Raheem, and V Sivakumar. Credit card business in malaysia: A data analytics approach. 12

[ZC18] Alice Zheng and Amanda Casari. *Feature engineering for machine learning: principles and techniques for data scientists.* " O'Reilly Media, Inc.", 2018. 7, 9, 10

[ZHL12] Bing Zhu, Changzheng He, and Panos Liatsis. A robust missing value imputation method for noisy data. *Applied Intelligence*, 36(1):61–74, 2012. 9

[ZLL+05] Yu Zhao, Bing Li, Xiu Li, Wenhuang Liu, and Shouju Ren. Customer churn prediction using improved one-class support vector machine. In *Proceedings of the International Conference on Advanced Data Mining and Applications*, pages 300–306. Springer, 2005. 14