NOVA IMS Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

Customer Churn Prediction in Insurance:
Modeling Renewal Price Elasticity of the Workers' Compensation Portfolio from Ocidental Seguros

Laura Sofia Sauthoff da Ponte e Castro

Internship report presented as partial requirement for obtaining the Master's degree in Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

CUSTOMER CHURN PREDICTION IN INSURANCE:

MODELING RENEWAL PRICE ELASTICITY OF THE WORKERS' COMPENSATION PORTFOLIO FROM OCIDENTAL SEGUROS

by

Laura Sofia Sauthoff da Ponte e Castro

Internship report presented as partial requirement for obtaining the Master's degree in Advanced Analytics

Advisor: Jorge Morais Mendes

Co Advisor*:* Nuno António

November 2021

ACKNOWLEDGMENTS

First, I would like to start to thank my advisor, Professor Jorge Morais Mendes, for being a part of this project since its beginning, supporting me in all the challenges that I have faced. As well, my co-advisor, Professor Nuno António, for helping me go further with his remarkable vision and insights. Not only by all the patience and support they have provided, but also, by all the knowledge they have passed during the lectures in the first year of the Master's degree.

Moreover, I want to express my gratitude to my internship supervisor, João Pedro Oliveira, for trusting me with this project and for all the support, knowledge, and opportunities he has given me for these months. Also, André Pousinho and Joana Negra, for all the experience they have shared with me in the first part of my internship, allowing me to understand the insurance market and, particularly, the Workers' Compensation branch.

I would also like to thank *Grupo Ageas Portugal* for allowing me to join them for one year, and everyone who has shared with me this path. Furthermore, to all the non-life branch Pricing and Business Analytics team who have made me feel so welcome and were always willing to help in anything I would need.

Lastly, I would like to thank my friends and family for believing in me and being a part of my academic journey and life.

ABSTRACT

Customer churn has been increasing in insurance, mainly due to technological improvements that allow customers to explore other insurance providers' offers. Given this, insurance providers need to compete among them, not only to get new customers but, to maintain their own.

This report results from a project developed during an internship in *Grupo Ageas Portugal*, which has different insurance brands such as *Ocidental Seguros*. This project's main goal was to model, with a monthly periodicity, customer churn of this latter's Workers' Compensation portfolio to improve the company's competitiveness and, ultimately, profit.

Many of the company's customer churn happens at their policy renewal time, where the only variable that the company detains control over is the price (premium) variation. Hence, by considering the premium variation and other relevant predictive variables, the goal was to predict the probability of a given customer to churn, allowing the company to optimize the current renewal's pricing process and maximize this branch's profit.

Thus, different variables that could influence the company's customer behavior were collected—one of those was the customer's location. Given the high dimensionality that such variable would represent and the small dataset available for modeling, clustering analysis is used to create new significant (with fewer dimensions) customer geographical areas.

Different supervised learning algorithms were then evaluated accordingly to their performance in predicting customer churn. The predictive models used were a Gradient Boosting, an Extreme Gradient Boosting, a Logistic Regression, and a Multilayer Perceptron. Given that the number of customers that renew their contracts is much superior to the number of customers who churn, Synthetic Minority Oversampling Technique (SMOTE) was used to create less unbalanced datasets (with synthetic samples) and evaluate the impact on the performance of one of the models.

Lastly, to guarantee a successful integration of the models into the renewal's pricing process, models were evaluated accordingly to the two business goals. First, by translating the observed evaluation metrics into profit. Secondly, by assuring that the customer's price elasticity would be captured, assuring a monotonic increasing relationship among the policy's premium variation and probability of churn.

KEYWORDS

Supervised Learning; Classification; Customer Churn Prediction; Non-Life Insurance; Renewal Price Elasticity; Clustering; Neural Network; Logistic Regression; Stochastic Gradient Boosting; Extreme Gradient Boosting

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AUC	Area Under the Curve
FPR	False Positive Rate
GB	Gradient Boosting
LR	Logistic Regression
MLP	Multilayer Perceptron
PFI	Permutation Feature Importance
ROC	Receiver Operating Characteristic
SMOTE	Synthetic Minority Oversampling Technique
TNR	True Negative Rate
TPR	True Positive Rate
XGBoost	Extreme Gradient Boosting Model

1. INTRODUCTION

This report results from a one-year internship in *Grupo Ageas Portugal*, one of the largest insurance groups in Portugal, holding the second-biggest market share. The internship was in the non-life branch's accidents team and, it is possible to divide it into two parts.

First, as a Pricing and Business Analytics team member, the goal was to support the team's usual business and monitoring analysis tasks. This part provided first insights about the business and, mainly, how the renewal's pricing process proceeds. The latter is a price (premium) adjustment done by the insurance company during contract renewal.

Secondly, the objective was to develop a project that could add value to the team's activities, particularly the renewal's pricing process. The project focused on the Workers' Compensation portfolio from *Ocidental Seguros* (one of *Grupo Ageas Portugal's* brands). The main goal was to design an analytical approach to compute the churn probability (given a particular price variation) at the time of each policy's contract renewal.

This report will only focus on the project developed in the second part of the internship. Thus, most of the first part's experience and learnings were crucial to the development of this seven-month project and, therefore, are here reflected.

1.1. PROBLEM STATEMENT AND OBJECTIVE

The price adjustment done to the potential renewals policies is called renewal's pricing process and is crucial to maintain the company's customers and the portfolio's profitability. This price adjustment depends not only on the policy's characteristics but also on the company's current situation: a balance between premiums, costs, and other market factors.

Renewal's pricing process core bases are the insurance principles of mutuality and solidarity. Given this, elements from the same risk class belong to a common fund and share damages costs, so that, if one of the fund's elements has unexpected costs, consequences can be minimal. Therefore, customers with unpleasant results, in terms of claims' costs, are not the only ones that might suffer an increase in their annual premium, but also profitable customers.

Naturally, these increases in the customer's premium lead many of them to cancel their contracts, especially given the ease of rapidly exploring other insurance providers' offers. Hence, given the high competitiveness of the insurance market, it is essential that insurance companies preserve their customers, not only focus on acquiring new ones.

Thus, having a model to predict the probability of a customer churning with a particular price variation supports the decisions made in the renewal's pricing process, ideally allowing a decrease in customer churn. The goal is that, given the customer's probability of churn to adjust the premium variation (obtained as an output of the process) and retain profitable customers that were likely to churn (with the given premium variation).

Although the policies renew their contracts yearly, policies have different start/renewal dates and, consequently, renew their policies at different moments in the year. Therefore, the renewal's pricing process happens with a monthly periodicity and, the constructed model should also deliver predictions

with this consistency. The purpose of running with this periodicity is to have the most recent information possible for each potential renewal policy. Some information, especially premium, suffers adjustments during the policy's annuity.

1.2. SUMMARY OF THE PROCESS

The CRSIP-DM (cross-industry process for data mining) methodology was applied in the execution of this project. This methodology suggests that the life-cycle of a data mining project can be broken into six phases that can be furtherly segmented into several tasks (Wirth & Hipp, 2000). Said that, at the top-level, it is possible to organize a data-mining project into the following six primary phases:

1. Business Understanding
2. Data Understanding
3. Data Preparation
4. Modeling
5. Evaluation
6. Deployment

It is not precisely necessary to perform the suggested tasks entirely in the proposed sequence. As the methodology suggests (Figure 1.1), moving back and forward is usually required. It is possible to find more details about the methodology in Wirth & Hipp (2000).

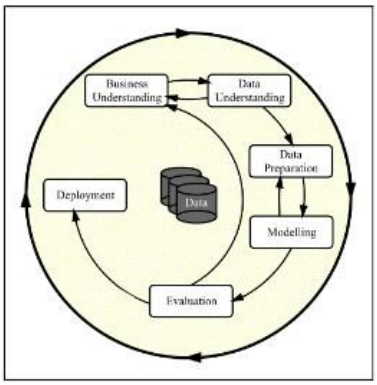


Figure 1.1 Phases of CRISP-DM (Wirth & Hipp, 2000)

Therefore, these main phases are covered in this report and further segmented into the methodology's different suggested tasks.

This report is organized as follows. Section 2 provides the theoretical framework - first, a customer churn and insurance churn modeling literature review; then, a theoretical review on machine learning theory and some of the used techniques. Section 3 provides an overview of the collected data for modeling the portfolio's customer churn. Next, in Section 4, the applied methodology and all its phases and tasks are explained. Section 5 presents the churn prediction models results and its discussion. Finally, Section 6 contains the conclusions from this project.

As for used software, SAS Enterprise Guide was used to collect the data, and Python was used in the remaining tasks.

2. THEORETICAL FRAMEWORK

2.1. LITERATURE REVIEW IN CHURN MODELING IN INSURANCE

Günther et al. (2014) define that the customer has churned *"When a customer cancels all his/her policies, either to switch insurance provider or because the need of insurance is no longer present*".

However, there is a differentiation in the types of customers' churns, and not all are suitable for analysis. A churner can be classified as voluntary when the customer deliberately decides to switch to another service provider or involuntary (usually not included for modeling) if circumstances like death or bankruptcy occur (X. Zhao et al., 2009).

Nowadays, customers can easily compare other insurance providers' offers in terms of premiums and coverages just using the internet. As a result, this transparency leads many customers to voluntarily switch to other insurance providers and cancel the insurance product they have on the company. This transparency has a significant impact on the insurance market's competitiveness.

Moreover, customer defections can have more to do with a service company's profits than many factors associated with a competitive advantage, given that profit can be boosted by almost 100% by retaining just 5% more of their customers (Sasser & Reichheld, 1990). Furthermore, attracting new customers can be twelve times more costly than retaining the company's current ones (Price, 2002).

Given this, especially now, to enhance competitiveness, companies should focus on reducing customer churn. Although Sasser & Reichheld (1990) are the grounders of the "zero defections" theory, they defend that companies should not try to eliminate all defections but be prepared to spot customers who leave and act accordingly to their findings.

Hence, different industries such as telecom, game, finance, and insurance use churn analysis to detect early churn signals and identify customers who are highly likely to leave voluntarily (Vafeiadis et al., 2015).

Although the goal among the industries is the same, the preferred models are different due to the differences in types and cycles of the log data. For the insurance sector case, the log data is relatively small. There are no significant changes in the customer's information, so many statistical approaches using traditional machine learning models or survival analysis are seen (Ahn et al., 2020).

Given this, a quick overview of the most commonly used techniques in the insurance sector was done. The most recent literature on the topic (last five years (2015-2020)) was selected for this analysis. Based on the results, the most used algorithms present in literature for insurance churn modeling are Logistic Regression, Decision Tree, Support Vector Machine, and Neural Networks. However, Gradient Boosting Model has shown the ability to attain good results, being one of the optimal models in Y. He et al. (2020).

Table 2.1 summarizes the result of the analysis.

Model	Used in Literature
Logistic Regression	Y. He et al. (2020) Vafeiadis et al. (2015) Sundarkumar & Ravi (2015) Bolancé et al. (2016) Spiteri & Azzopardi (2018)
Decision Tree	Vafeiadis et al. (2015) Sundarkumar & Ravi, 2015) Bolancé et al. (2016) Dolatabadi et al. (2017) Spiteri & Azzopardi (2018) Scriney et al. (2020)
Support Vector Machine	Y. He et al., (2020) Vafeiadis et al. (2015) Sundarkumar & Ravi (2015) Bolancé et al. (2016) Dolatabadi et al. (2017) Spiteri & Azzopardi (2018) Scriney et al. (2020)
Neural Network	Y. He et al. (2020) Vafeiadis et al. (2015) Sundarkumar & Ravi (2015) Bolancé et al. (2016) Dolatabadi et al. (2017) Scriney et al. (2020)
Gradient Boosting	Y. He et al. (2020)
Naïve Bayes Classifier	Vafeiadis et al. (2015) Dolatabadi et al. (2017) Spiteri & Azzopardi (2018) Scriney et al. (2020)
Random Forest	Y. He et al. (2020) Spiteri & Azzopardi (2018)
Extra Trees Classifier	Y. He et al. (2020)

Table 2.1 Models used in literature for insurance churn modeling

2.2. MACHINE LEARNING

Unsupervised Learning

Unsupervised learning is a form of machine learning that aims to group data into segments based on similar attributes, or naturally occurring trends, patterns, and relationships hidden in the data (McCue, 2015). Standard algorithms used for this kind of task are clustering, anomaly detection, neural networks, and approaches for learning latent variable models (El Bouchefry & De Souza, 2020).

The main goal of clustering analysis is to segment the finite unlabeled dataset into a finite and discrete set of hidden data structures (Xu & Wunsch, 2005). The result of this analysis is several groups of the data named clusters. These are subsets of data grouped when, according to the studied criteria, have similar characteristics and, when not, separated into different groups (Rokach & Maimon, 2005).

Moreover, two types of clustering techniques are partitional clustering and hierarchical clustering. In the first, groups are created by partitioning the space into a pre-defined number of subspaces. On the other hand, in hierarchical clustering, the data objects are grouped in sequence by a hierarchical structure.

Supervised Learning

Another form of Machine Learning is supervised learning. Contrary to unsupervised learning, it uses a dataset that has already been classified (labeled) as a basis for predicting the classification of other unlabeled data (Talabis et al., 2015). The labeled dataset is a training set composed of input variables (features) and an output variable (label). The used features will influence the model's ability to correctly classify the predicted variable, the output variable (L. Wang et al., 2021).

Supervised Learning can be divided into classification when the label is categorical and regression when the label is continuous. Therefore, the studied problem is a classification task since churn modeling can be translated to a binary classification problem: assuming 1 when the customer churns and 0 when the customer does not churn.

2.2.1. Data Imbalance

For many real supervised learning problems involving a binary response variable, datasets present a skewed distribution, having one class with a much lower representation than another. When the dataset shows this type of behavior, having an underrepresented class, the data is said to be unbalanced (S. Wang & X. Yao, 2012). Researchers have concluded that this imbalance causes a suboptimal classification performance (Chawla et al., 2004), since classifiers tend to give much higher importance to the larger classes.

H. He & Garcia (2009) bring that. Usually, classifiers, when dealing with an imbalanced dataset, tend to "*provide a severely imbalanced degree of accuracy, with the majority class having close to 100 percent accuracy and the minority class having accuracies of 0-10 percent*", such consequence can represent high costs to some industries so, therefore, is vital to construct a model that "*will provide high accuracy for the minority class without severely jeopardizing the accuracy of the majority class*".

In order to deal with imbalanced datasets, three possible approaches can be taken: data level, algorithmic level, and combining or ensemble methods, for which the first involves resampling to reduce the class skewness (Yap et al., 2014). Resampling is done either by removing instances from the

majority class (undersampling) or adding instances to the minority class (oversampling) by using algorithms such as SMOTE.

Synthetic Minority Oversampling Technique

Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002) is one of the most popular and influential data pre-processing algorithms to deal with the data imbalance problem (García et al., 2016).

This technique is an oversampling approach, said that new instances from the smaller class are introduced into the dataset. Unlike basic approaches such as random oversampling (ROS), which only duplicates samples from the minority class, SMOTE generates new synthetic samples, overcoming the overfitting caused by approaches like ROS (Fernández et al., 2018).

The first step of this technique is to define the amount of oversampling. Here, it is possible to either set up this value to approximate a balanced class distribution or discover it via a wrapper process (Chawla et al., 2008). Then, based on k nearest neighbors and linear interpolation ideas, the synthetic samples are created. SMOTE operates in the feature space rather than in the data space, each minority class sample is considered along with its k nearest neighbors, and the new samples are introduced along the line segments joining them (considering any/all of the k neighbors) (Chawla et al., 2002).

2.2.1.1. Model Calibration

When using such techniques of artificially rebalancing the dataset or even by consequence of the dataset's characteristics, the training and test sets have different distributions. This difference in training and test sets distribution violates the basic assumption in machine learning that both are drawn from the same underlying distribution (Pozzolo et al., 2015). By violating this assumption, the predictions obtained in the test set will be biased and, therefore, enhance the need for probability calibration to obtain unbiased predictions.

Furthermore, some methods tend to bias predicted probabilities by pushing away or closer to 0 and 1, enhancing the need for calibration (Niculescu-Mizil & Caruana, 2005). Dormann (2020) even states that "*it should be applied to any model type as part of the prediction process, before predicting, cross-validating and making effect plots and maps or using predictions in any other probabilistic interpretation*".

Two model calibration methods that can be used to correct these biased probabilities are Platt Scaling and Isotonic Regression. The first is more effective when the distortion is sigmoid-shaped, and the latter is a more robust method that can correct any monotonic distortion but, more prone to overfitting (Niculescu-Mizil & Caruana, 2005).

The Platt Calibration calibrates probabilities by passing the output through a sigmoid:

$$P(y = 1|f) = \frac{1}{1 + e^{(Af+B)}} \quad (1)$$

Where $f(x)$ is the learning method and parameters A and B are estimated using Gradient Descendent, such that they are a solution to the following minimization function:

$$argmin_{A,B}\left\{-\sum_i y_i \log(p_i) + (1-y_i)\log(1-p_i)\right\} \quad (2)$$

where,

$$p_i = \frac{1}{1+e^{(Af_i+B)}} \quad (3)$$

On the other hand, the Isotonic Calibration is more general given that the only restriction is that the mapping function is isotonic (Niculescu-Mizil & Caruana, 2005). The basic assumption of Isotonic Regression (on which the model is based) is that:

$$y_i = m(f_i) + \epsilon_i \quad (4)$$

Where, y_i are the true labels, f_i the model's predictions, m a monotonic increasing (isotonic) function. The goal is to find $\hat{m}$, using the true labels and model's predictions as a training set, such that:

$$\hat{m} = argmin_z \sum (y_i - z(f_i))^2 \quad (5)$$

One algorithm that can be used to find a stepwise constant solution for this problem is the pair-adjacent violators (PAV) algorithm (Ayer et al., 1955).

Lastly, one way to assess how well-calibrated the model is can be through a calibration plot. Here, "*a set of predictions of a binary outcome is well calibrated if the outcomes predicted to occur with probability p do occur about p fraction of the time, for each probability p that is predicted"* (Naeini et al., 2015), which can be translated into a straight line from (0,0) to (1,1). Given this, in the calibration plot, the x-axis represents the average predicted probability in each bin. The y-axis represents the observed fraction of samples in the bin whose real labels are positive. The observed curve is then compared with the straight line $y = x$.

2.2.1.2. Threshold-Moving Method

A technique that should be considered when dealing with class imbalance is changing the decision threshold (model's continuous output cut-off) and adapting it to a performance metric. The main difference between rebalancing (using techniques like SMOTE) and threshold-based methods is that the latter relies on manipulating the continuous output of a learned model instead of relying on data pre-processing before the learning happens (Collell et al., 2018).

Provost (2008) even states that "*The bottom line is that when studying problems with imbalanced data, using the classifiers produced by standard machine learning algorithms without adjusting the output threshold may well be a critical mistake*".

The threshold moving method uses the original training set to train and tunes or shifts the decision threshold by adapting it to a performance metric. One possible approach is to use the ROC evaluation procedure and move from where misclassifications attain their maximum on the positive class to the point where the maximum in the negative class is attained, selecting the point where the curve attains its maximum (H. He & Garcia, 2009).

2.2.2. Feature Selection

A well-known problem is the "curse of finite sample size", for which the relationship among the number of samples available and the features considered for modeling needs to be considered (Jain & Chandrasekaran, 1982). Each new feature introduced to the model will represent a new dimension. The higher the dimensionality, the sparser the dataset becomes and, thus, lower the feature space coverage (Verleysen & François, 2005). Consequently, the problem's complexity rapidly grows with the introduction of more dimensions. Bellman (1966) introduced the term Curse of Dimensionality to explain such phenomena.

Therefore, and given the nowadays existing high-dimensional data, feature selection is one of the essential techniques in data preprocessing by eliminating irrelevant, redundant, or noisy features (Kalousis et al., 2007). Performing such techniques allows faster algorithms and, besides improving predictive power, also improves comprehensibility (Kumar & Minz, 2014).

It is possible to broadly classify feature selection methods into filter and wrapper methods, where the first ranks features based on statistical measures, independently of the learning algorithm (Kohavi & John, 1997). One example of this type of method is to use as importance measure (score) the variable's correlation with the target. On the other hand, wrappers evaluate each candidate subset of features' impact on a particular learning algorithm. The latter approach usually allows achieving better results given their close interaction with the classifier (El Aboudi & Benhlima, 2016).

Some examples of wrapper methods are forward selection, backward elimination, and recursive feature elimination. The first keeps adding new features which improve the model performance until no other feature respects the criteria. The second works similarly but oppositely, starts with all features, and removes the least significant effect that does not meet the model's staying criteria until all features are significant. In both (forward and backward elimination), the feature stays in the model once added/removed. Lastly, recursive feature elimination is similar to the forward selection method. In this case, effects are added and removed into the model such that one or more backward elimination steps can happen after a forward selection step (Bursac et al., 2008).

Another family derived from the two previous method families (filter and wrappers) are embedded methods. These methods combine the classifier development with the search of the optimal subset of features, capturing dependencies at a lower computational cost than wrappers but (like wrappers) also have a risk of overfitting (Seijo-Pardo et al., 2017). Examples of embedded methods are Lasso and Ridge regression, which have built-in feature selection methods that employ L1 and L2 regularization (respectively).

2.2.2.1. Ensemble Learning for Feature Selection

Ensemble Learning is a type of learning where multiple models are trained and combined to solve the same problem (Polikar, 2006). Ensemble Learning is based on the assumption that combining the solution of multiple experts is better than using the solution of a single one. In this way, a set of hypotheses is constructed rather than using only one single hypothesis to explain the data, making it possible to reduce bias and variance from the learning algorithms (Dietterich, 2002).

Although usually employed to improve classification results, it is also possible to use ensemble learning as a feature selection technique. Combining multiple feature selection methods (instead of relying on

just one) makes it possible to attain more robust feature subsets, showing a great promise to high-dimensional datasets with small sample sizes (Saeys et al., 2008).

The approach benefits from diversity and control of variance and is possible to employ in two ways: one is to use the same algorithm to retrieve the features' importance using different subsets of the data (data perturbation / homogeneous) and, other, is to use different feature selection techniques in the same dataset (function perturbation / heterogenous) (Chiew et al., 2019).

Said this, different levels can be varied and shall be chosen when employing ensemble learning in feature selection, following Bolón-Canedo & Alonso-Betanzos (2019) can be defined as follows:

- Dataset Level: use different subsets of data (or not)
- Feature Level: use different subsets of features (or not)
- Learner Method Level: use or design different learning algorithms (or not)
- Combination Level: Use or design different combination/aggregation methods
- Threshold Level: use of different thresholding methods (in case of using ranker methods)

Bolón-Canedo & Alonso-Betanzos (2019) also verified that heterogeneous feature selection ensembles are more commonly used than homogeneous ones. However, it is possible to obtain either a feature subset or a feature ranking in both cases, depending on the type of feature selectors. For the latter, a threshold method needs to be defined.

When using feature rankers, several feature ranking algorithms of the ensemble are combined, creating a final ranked list of the features, given the features' relevance to prediction. The approach for combining the ensemble members' results (ranks) has different proposals in the literature, from simpler to more complex solutions (Seijo-Pardo et al., 2015). Some of the most popular straightforward methods to combine such ranks attributed to each feature are: minimum (best) rank, median rank, arithmetic mean rank, and geometric mean (Bolón-Canedo & Alonso-Betanzos, 2019).

Given the threshold method decision, the most common approach is defining a fixed percentage of the top features, but this percentage depends on the used dataset (Bolón-Canedo & Alonso-Betanzos, 2019). Therefore, this technique is not optimal since it is prone to overstating or understating this cut-off value (Chiew et al., 2019).

Permutation feature importance

Permutation feature importance (PFI) is a model-agnostic feature selection method. Therefore, it is possible to perform a heterogeneous approach by using this algorithm as a base learner for the ensemble and introduce variability by using different base models in the algorithm.

This permutation feature importance measurement was firstly introduced for random forests in 2001 (Breiman, 2001) but can be used in any model. PFI assesses the variable's importance for the given model when its relationship with the target is broken by observing the decrease in the model's score when random noise replaces a variable (introduced by randomly shuffling the variable's values) (McGovern et al., 2019). Therefore, it is possible to understand how important a given feature is to the model's ability to predict the target correctly.

PFI's operation method makes it sensible to correlated features (Strobl et al., 2008) and is, therefore, essential to address this issue primarily. However, PFI has advantages such as robustness not to bias the measures favoring high cardinally features over binary features.

3. DATA

Given the small dimensions of the Workers' Compensation portfolio, the main goal in the data collection phase was to extract as much data as possible. Moreover, to guarantee that the collected information truly captures the reality, a 3-month aging is required.

Therefore, the data extraction process was divided into two phases. A first phase, at the project's beginning, in which it was possible to extract data regarding the renewals and churns from January 2017 until September 2020, used for the models' construction. Then, a second phase, during the Model's Assessment phase, to be used as a test set. This new dataset contained data from October 2020 until February 2021, including January, where most of the policies renew their contracts. Bellow, the Data Extraction phases can be seen in Figure 3.1.

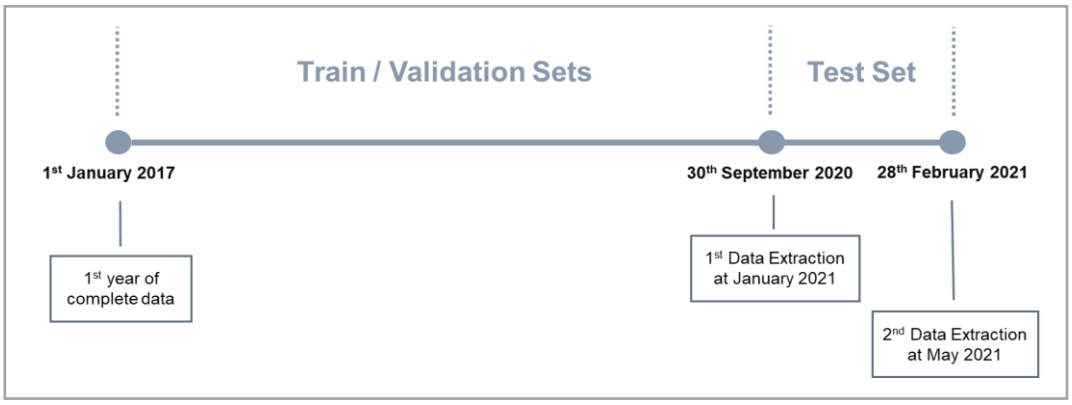


Figure 3.1 Different data extraction phases

Given the problem context, the goal was to identify voluntary churners who based their decision on price variation. In this way, neither involuntary churns nor cancelations outside the defined renewal window (explained in section 4.1) are included in the analysis.

Moreover, to guarantee that the information extracted would capture the event that was decided to be studied - voluntary churners that have churned by consequence of the premium variation - some restrictions have been applied to the data:

- Not include churns that are a consequence of, for example, bankrupt of the company.
- Not include policies from employees of the company
- Not include temporary policies
- Not include the group of policies flagged by the company that receive different renewal's pricing processes from remaining customers (customers with higher total premiums and policies identified to go through a pruning process)
- Not include policies that cancel or issue their exit before the defined renewal window
- Not include policies for which it was not possible to extract their premium value
- Not include annuities with a duration of less than one year

Policies that have canceled their contracts to acquire a new policy in the company (policy cannibalization) were still considered for analysis, since it is possible that the customer has cancelled due to the price increase. Usually, customers acquire a new policy to obtain a lower price maintaining the same benefits.

With such constraints applied, it was possible to extract a dataset with the policies' annuity information before the renewal date in which, for one year, a particular policy only appears once. Regarding the collected variables, this decision was made considering the information present in the literature and suggestions made by the project's stakeholders. The collected variables and their description were not included in this report. However, it is possible to resume the variable's information into the following categories:

- Category 1: Value paid for insurance
- Category 2: Customer behavior
- Category 3: Customer awareness of the increase
- Category 4: Customer's/Product's characteristics
- Category 6: Customer's claims/cost and its management by the company
- Category 6: Position of *Ocidental Seguros* in Market
- Category 7: Customer's interaction channel with the company
- Category 8: Customer's loyalty
- Category 9: Customer's geographical location characteristics
- Category 10: Pandemic situation
- Category 11: Discounts

Given that the model needs to deliver predictions 60 days before the policy's renewal date (also explained in section 4.1), all the variables collected had to consider this vision on time. Most variables maintain constant along with the annuity. However, others, such as the employees' total salary associated with the contract, usually vary during this period. Therefore, the extracted variables were designed to gather the correct vision, with the end of annuity standing for the 60 days before it.

As mentioned before, the most crucial variable for measuring its impact on churn is the premium variation obtained in the renewal's pricing process. However, this premium variation also depends on the contract's employees' total salary, which may vary from one annuity to another or even in the 60 days window. In this way, it was decided that the premium variation should only be measured, excluding the variation of this variable. Therefore, only the contract's tariff variation was considered when measuring the premium variation.

4. METHODOLOGY

4.1. BUSINESS UNDERSTANDING

As mentioned before, this project is applied to the Workers' Compensation portfolio from *Ocidental Seguros*. To guarantee the project's success, the first objective was to understand this business line and the project objectives. Further, this knowledge has been converted into data mining goals, and a project plan has been designed.

Workers' Compensation insurance has been mandatory in Portugal since 1993 for Third-Party companies' workers and extended for self-employed workers in 1997. This type of insurance has only one coverage that covers the risk of accidents in the employees' workplace or on their way from/to home. This coverage assures the legally needed benefits by consequence of any of the involved employees' accidents, covering all needed expenses to ensure the worker's total recovery (or compensation in case of disability or death). The paid premium for this insurance depends not only on the total secured employees' salary (sum insured), which may vary during the year but also on the contract's rate stipulated by the company (which can vary yearly in the renewal's pricing strategy).

The renewal's pricing process proceeds approximately two months in advance (given the customer's renewal date). By law, the insurance company must send the renewal letter informing the customer about the next annuity's premium 30 days in advance. Given this, the market defined that this renewal letter should be sent 45 days in advance, so this is (usually) when the customer is aware of the variation in the premium (contract's rate). Therefore, based on this business knowledge, it was defined that only a cancellation that happens in a window of 45 days prior and 50 days after the end of the policy's renewal date would be classified as churn. Moreover, the annuity's premium variation is determined 60 to 45 days before the end of the policy's renewal date, so all variables extracted respect this time window (using the variable's vision 60 days before the policy annuity end date).

Ocidental Seguros has a bancassurance channel and detains 1.5% of the Workers' Compensation insurance market share. The company's portfolio can be segmented into three main customer groups:

- **Housekeepers** – Besides housekeepers, which is the main component of this group, other domestic employees are integrated, as gardeners, for example.
- **Self-Employees** – In general, are small companies where the insured entity is the worker himself.
- **Third-Party** - Comprises companies from diverse dimensions (micro/small, medium, and big) insuring their employees.

Although most of *Ocidental Seguros'* customer contracts are from the Housekeepers' segment, Third-Party represents almost 85% of the portfolio's total annual premium. Such behavior is explained by the fact that, on average, the total secured employees' salary for this segment is much higher than for the other segments.

Regarding claims, the most significant frequency of claims is observed for the Self-Employees' segment. However, was the Third-Party segment for which, at the time, the lowest profitability was observed.

4.2. DATA UNDERSTANDING

Once having the first insights on the business, data was collected. Here, the different variables were extracted respecting the conditions mentioned above, data was merged from several sources, and most inconsistencies were directly approached. These described tasks, all performed in SAS Enterprise Guide, ended up being one of the most time-consuming parts of the project.

After having the data available, a data exploration was carried on having the first insights on the data, discover possible data quality issues and activities that need to be dealt with in the Data Preparation phase. These tasks were also crucial to secure a robust project plan in the Business Understanding phase.

While performing the data exploration, it was observed that the Self-Employees, Housekeepers, and Third-Party segments presented different characteristics and behaviors. They suffered, as well, distinct renewal's pricing strategies since the company's business goals are different for each of the customer segments.

Therefore, it was decided that, in the problem context, it would make sense to separate them and construct three different models (one for each customer segment). Bellow, in Table 4.1, it is possible to observe the dataset's division along with the observed churn rate.

Customer Segment	Sample Size	Churn Rate
Self-Employees	12.797	8.7%
Housekeepers	33.501	5.4%
Third-Party	22.637	9.1%

Table 4.1 Data aggregation for modeling

The different characteristics and behaviors of the segments also impact the relevance of the observed variables. For example, by segmenting the data, some of the gathered variables' information was no longer valuable and were, therefore, directly excluded.

4.3. DATA PREPARATION

The data preparation phase covered all activities from the initial raw data to the final dataset used for modeling. The tasks managed in this phase can be separated into Data Selection, Data Cleansing, Feature Engineering, and Data integration and Format. These tasks are described below.

4.3.1. Data Selection

By performing some statistical analysis and visualizations, some information was considered to be irrelevant. Subsequently, features that presented only or mainly one single value (with low variance) were considered irrelevant, having no (or not enough) predictability power for the given customer segment and, therefore, excluded.

Redundant information was present as well. Correlated features unnecessarily increase the feature space and can impact the model interpretation, masking meaningful interactions, and therefore, essential features appear irrelevant (Tološi & Lengauer, 2011). Therefore, such features were identified and eliminated, avoiding such negative impacts in processes like feature selection.

Although the Pearson correlation coefficient is the most common measure of correlation, the correlation measure used was Spearman's rank correlation coefficient, which measures the monotonic relationship between the two variables. Although this latter is calculated similarly to Pearson's, Spearman's rank correlation coefficient ranks both x and y to transform them to values between 1 and N, allowing this coefficient to be relatively robust to heavily tailed distributions and outliers (De Winter et al., 2016).

Said this, all groups of features presenting a Spearman's rank correlation higher than 0.6 were analyzed. With the acquired business knowledge, correlations were eliminated by removing the less relevant feature.

4.3.2. Data Cleansing

Data Cleansing is the process of improving the data quality in an existing system and may be directly tied to data acquisition (Maletic & Marcus, 2000). In this project, this was as well the approach. Firstly, as mentioned above, inconsistencies found on the data by performing some designed validations tests were already dealt with within the data acquisition process. Then, two other essential tasks to guarantee data quality were performed: outlier detection and handling missing data.

As for outlier detection, a more conservative approach was taken since such values can represent individuals with different behaviors in a normal situation and possibly contain helpful information for the models. Consequently, an external dataset without instances considered to be extreme values was created to understand if such elimination helps the models.

Finally, the missing data were handled. Given that the missing data present in the dataset is not considered NMAR (not missing at random) (Rubin, 1976), missing values imputation methods can be considered. Replacing the missing values with a new value is usually preferred over eliminating the variable with missing values or even the row, keeping the maximum information possible. However, supposing that the number of missing values in a variable is too high, not dropping such variable might lead to the risk of providing misleading information to the model given the high number of synthetic values. Therefore, the approach used to deal with the missing values varied accordingly with each variable's characteristics. Moreover, a threshold was defined, to consider a variable's removal depending on the variable's significance.

Missing Values in Numerical Variables

For numerical variables with missing values was decided to use the other variables present in the dataset and take advantage of their relations with the variables. These relations can be captured by computing the correlations among the variables with missing values and other variables present in the dataset and predicting the missing value's value. Constructing a prediction model can maintain those relations, avoiding more straightforward techniques like mean imputation, which reduces variance in the dataset. Here, one widely used machine learning technique for missing data imputation, the K-nearest neighbors (Batista & Monard, 2003), was used.

Missing Values in Categorical Variables

Regarding the missing values found on categorical variables, a special value characterizing the information's absence already existed for some cases, so this value was assigned to the missing ones. For other variables, for which the percentage of missing values was relatively low (<1%), the mode was

inputted. A more straightforward method would not introduce a significant bias given its low exposure. At last, according to the acquired business knowledge, other categorical variables with missing values were highly related to the customer's location. Therefore, the customer's municipality mode was introduced, always tracking this imputation's impact on the final variable's distribution.

4.3.3. Feature Engineering

As a result of the previous tasks, new attributes are created derived from the dataset's existing features. Also, in this task, generating new records to assist the modeling phase and improve the model's performance on the dataset is considered.

4.3.3.1. Derived Attributes

Most of this work was carried on already in the data collection phase. Some features were combined to create more value to the studied problem. However, a significant portion of time was used on the collected features regarding the customer's location. Here, the main goal was to create a new variable to identify the customer's geographical location considering the observed similarities/dissimilarities across "neighbors".

The data collection task made it possible to obtain the customer's seven-digit zip code, consequently accessing their district and municipality. The main problem with these variables was that even considering the highest level of aggregation (customer's district), the geographical location would still be represented by a categorical variable with 18 different values, which significantly increases the dimensionality of the feature space. As shown before, limiting the number of features considered for modeling is crucial, especially for small sample sizes. Besides this, the customer's distribution throughout the different locations is significantly different, given that some of them have a low exposure of customers and, consequently, a possible low predictability capacity.

Said that, and given the business relevance of the customer's location, it was decided to group the customers' municipalities creating new areas with similar characteristics. Thus, the external demographic data about the Portuguese municipalities (that the company had available) was used, where correlated variables were removed for the analysis. The approach was to group all municipalities (more than 300) into similar areas in those metrics, with, therefore, possibly similar behaviors. Unfortunately, such information was not available for the Portuguese islands, so these were all together considered a different cluster since they did not have enough exposure to consider the separation of, at least, Madeira and Azores.

As shown before, to attain such a goal, clustering analysis can be performed. This unsupervised learning technique reduces the complexity of the data into groups of objects that are similar to each other. For this, both a partitional clustering and hierarchical clustering technique were attempted. As for partitional clustering, the K-means method was used, one of the most used unsupervised learning techniques (Pham et al., 2005). By the other hand, hierarchical clustering can either start with one observation per cluster and stop when all instances are in the same group (agglomerative) or start with all observations in the same group and sequentially separate them until all instances are in an individual group (divisive). Here, the Agglomerative Hierarchical Clustering was used.

Moreover, three approaches were necessary to lead to the final result and are described below. For each case, the clustering technique which held the best results was selected.

New Geographical Variable – First Approach

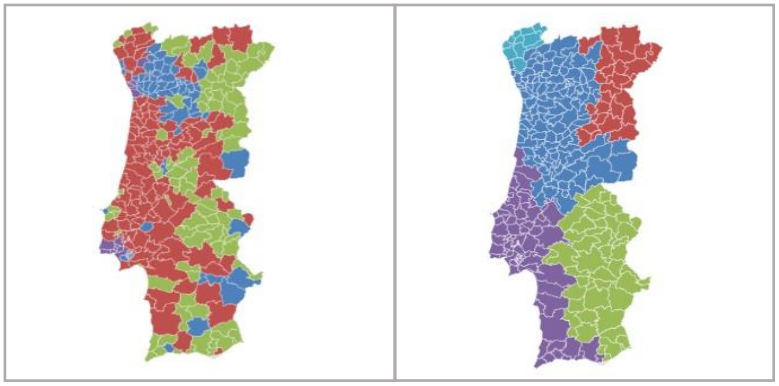
The first approach only used the external demographic data to explain the customer's behavior in the problem context. Both algorithms were applied, for which the best results, presented in Figure 4.1(a), were obtained using the K-means algorithm with $k = 4$.

However, the main goal is to construct a solution that is an alternative to the Portuguese districts and, as well, an appropriate new predictor for the model. One of the problems with the constructed solution was the lack of differentiability in the churn rate among the created geographical areas. The way the municipalities are grouped does not show a great dataset's separability when segmenting the information and observing the churn's behavior. Also, due to the first law of geography, there is spatial dependence and (positive) autocorrelation, where there is a higher relation among near events (Harris et al., 2005). Therefore, as a districts' alternative, these areas shall be contiguous, which was not the case in this solution.

New Geographical Variable – Second Approach

Therefore, in the second approach, both these problems are addressed. In order to try to create contiguous areas, each municipality's latitude and longitude were added to the analysis. Also, the percentage of churn observed for the given municipality in the last three years was included, so municipalities with high/low churn rates are differentiated. For this latter, if there were not enough policies in a given municipality, the district's churn percentage was considered for the municipality; otherwise, the measure could not capture the actual reality. However, since several municipalities did not have enough exposure, several municipalities presented the same percentage of churn, which could negatively impact the analysis. Given this, such municipalities were excluded from the clustering analysis and only later classified when the cluster centroids were already defined.

Moreover, like the first approach, all variables are scaled into a 0-1 scale to have the same importance/impact in the analysis. However, given that the main goal was to obtain contiguous areas that presented differences in their churn rate, a higher weight has been attributed to the three new variables (the 0-1 scale is multiplied by a constant value). Hence, given the bigger scale, dissimilarities in these variables have a higher penalization and, consequently, a higher impact on the final solution. At last, if the municipalities groups did not guarantee the contiguous areas, those would be reallocated to the area that made more sense in the analysis context. Figure 4.1(b) presents the analysis results, for which the agglomerative hierarchical clustering with $k = 5$, was chosen.



(a) **(b)**

Figure 4.1 Municipalities' aggregation in the two attempts. **(a)** Results of the first performed clustering analysis. **(b)** Results of the second clustering analysis performed

Although these results were more appropriate than the first, some modifications were still necessary. When the data was separated into the different customer segments, some problems had arisen, mainly for Self-Employees' customer segment (which had the smallest dataset). By doing this classification using the entire dataset's information, the created areas were no longer meaningful when segmented, this is, the differentiability in the target variable was no longer observed.

New Geographical Variable – Third Approach

As the third and final approach was decided that the second approach's strategy should be performed but, this time, segmenting the data before its application. Thus, although with some differences, three different contiguous areas were created for each of the three customer segments: "North Coast", "South Coast", and "Inland". Also, an additional area containing the Portuguese islands. This solution allowed to attain all the previously defined goals, maintaining a difference, for all customer segments, of almost one p.p. among the three different created areas' churn rate. In Figure 4.2 is possible to observe the representation of the new geographical variable created for each segment. For all segments, the inland had the lowest exposure, and the south coast had the highest.

Self-Employees' Geographical Areas

The Self-Employees' segment results are presented in Figure 4.2(a), and the created areas can be described as follows:

- **South Coast** – Geographical Area with the highest consumption index, higher average salary, and higher number of insurance agents.
- **North Coast** - Geographical Area with lowest consumption index and lowest number of unemployed. Highest average churn rate (9.5%).
- **Inland** - Geographical Area with the highest number of unemployed and lowest average salary. Lowest average churn rate (7.3%).

Housekeepers' Geographical Areas Description

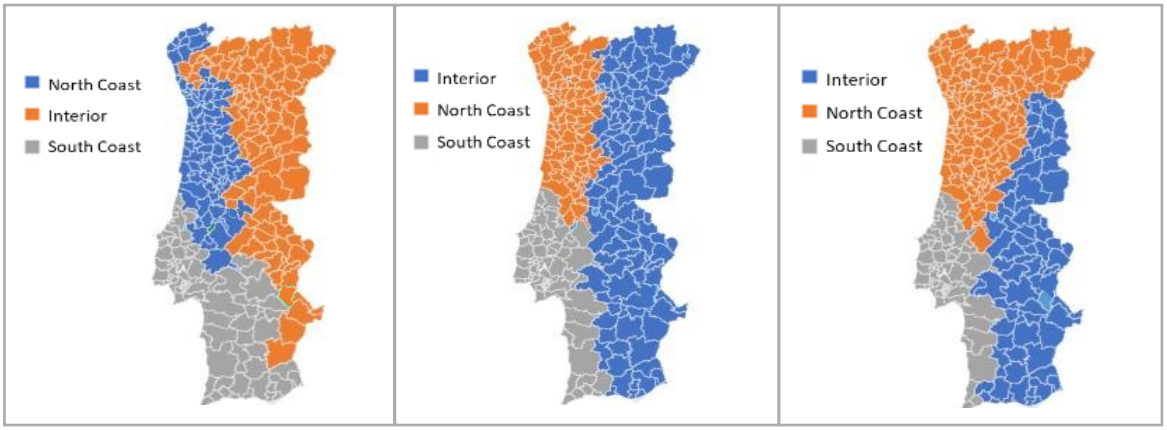
The Housekeepers' segment results are presented in Figure 4.2(b), and the created areas can be described as follows:

- **South Coast** – Geographical Area with the lowest number of unemployed, higher average salary, and higher number of insurance agents. Lowest average churn rate (4.9%).
- **North Coast** - Geographical Area with lowest consumption index.
- **Inland** - Geographical Area with the lowest average salary and higher number of unemployed, presenting a smaller number of insurance agents. Highest average churn rate (7.7%).

Third-Party's Geographical Areas Description

The Third-Party's segment results are presented in Figure 4.2(c), and the created areas can be described as follows:

- **South Coast** – Geographical Area with the lowest number of unemployed, higher average salary, and higher number of insurance agents. Lowest average churn rate (8.2%).
- **North Coast** - Geographical Area with lowest consumption index and highest average churn rate (10.5%).
- **Inland** - Geographical Area with the highest consumption index presenting the lowest average salary and fewer insurance agents.



(a) **(b)** **(c)**

Figure 4.2 Clustering analysis final results for each of the client segments. **(a)** Self-Employees' customer segment. **(b)** Housekeepers' customer segment. **(c)** Third-Party's customer segment.

4.3.4. Data Integration and Format

In this phase, newly created features are introduced into the dataset. Moreover, variables are transformed in a way that modeling requirements are respected. For example, in place of categorical features, $n-1$ dummy variables are created, where n represents the number of categories.

Besides this, variables can also be transformed to ease the process of feature space understanding. It is shown that, in reality, improving the variables' normality, especially when substantial non-normality is present, benefits almost all analyses; however, one of the downsides is the loss of the variable's interpretability (Osborne, 2010). Therefore, some variables that have shown a great right/left skewness were transformed to approximate the normal distribution better. In Figure 4.3, it is possible to observe one of the variable's transformations, comparing its original distribution with Yeo-Johnson transformation.

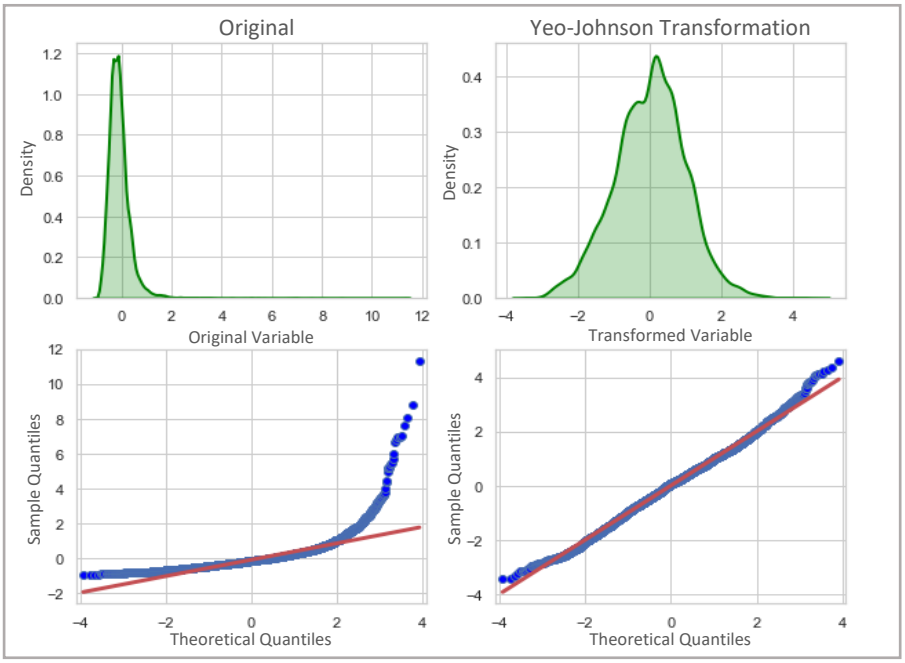


Figure 4.3 Example of a variable's transformation using Yeo-Johnson transformation - original vs transformed density plots and Q-Q plots

Yeo-Johnson transformation is a power transformation family with similar properties as Box-Cox transformation, but well defined in the whole real line, thus appropriate to reduce skewness and approximate normality without such limitation (Yeo, 2000). Box-Cox represents a family of power transformations (like square root, log, or inverse transformations - which are considered a way to meet the normality assumption) that easily find the optimal power transformation for a given variable to stabilize its variance (T. Zhang & Yang, 2017). Nevertheless, the original Box-Cox (Box & Cox, 1964) transformation is only valid for a positive x, contrary to Yeo-Johnson transformation.

On the other hand, it has also been shown that Feature Scaling generally improves the performance of classification algorithms (Bollegala, 2017). What generally leads to such improvement is that features' values, naturally, occupy different ranges (some ranging in thousands, others in tens) and, features with higher ranges tend to have a more decisive role while training the model. Furthermore, the feature's relative value difference is often more informative than its absolute value (Bollegala, 2017). Therefore, other than using the data in its original range form, some scaling methods were attempted. Such transformations were performed by estimating the scaling parameters using the training set and, the feature scaling method is then applied in both training and test sets.

4.4. MODELING

In this phase, the used modeling techniques are selected. Also, a test design is created to build the different models and correctly assess them, understanding their value for resolving the problem at hand.

4.4.1. Model Selection

Initially, for this project, four predictive models were selected and constructed: a Logistic Regression, a Multilayer Perceptron, a Gradient Boosting model, and an Extreme Gradient Boosting model. These methods are briefly explained below.

Logistic Regression
Similar to linear regression, logistic regression (LR) can include one or multiple covariates (variables) that, jointly with unknown parameters estimated from the data, produce a linear (and continuous) predictor. The logit function is used to guarantee that the outcome variable falls in a 0-1 range. Furthermore, it is possible to add a regularization term to the logistic regression, encouraging the fitted parameters to be small and helping prevent overfitting.

Multilayer Perceptron
The Multilayer Perceptron (MLP) is a feed-forward network composed of input neurons, output neurons, and hidden layers of neurons between them. Neural networks are a branch of artificial intelligence inspired in human brains. Here, numerous cells called neurons process information in parallel, linked together in a network by synapses, where intelligence is argued to be encoded. Moreover, in a feed-forward network, information only moves forward, from the input nodes to the hidden nodes and finally output nodes (Zell, 1994), connected by weights and output signals. A nonlinear transfer function modifies the neuron's weighted inputs, also called the activation function (Njikam & Zhao, 2016). These superpositions of nonlinear transfer functions allow the multilayer perceptron to approximate highly non-linear functions (contrary to the logistic regression) and accurately generalize when presented with new, unseen data (Gardner & Dorling, 1998).

Gradient Boosting Model

The Gradient Boosting (GB), proposed by Friedman (2001, 2002), belongs to the family of boosting methods, a type of ensemble learning. In boosting, a new model is added to the ensemble sequence trained based on the error of the whole ensemble learned so far (Natekin & Knoll, 2013). Hence, misclassified instances are emphasized by receiving higher weights in the next iteration, which, for example, usually happens to instances near the decision boundary (Meir & Rätsch, 2003). These weights represent the importance that the given instance will have in the following base learner construction. Therefore, instances that the previous base models have shown difficulty understanding appear more often in the training data (Y. Zhang & Haghani, 2015). The final model obtained by the boosting algorithm will be a linear combination of the several base-learners considering their performance on the dataset (G. Wang et al., 2011).

Extreme Gradient Boosting Model

The Extreme Gradient Boosting model (XGBoost) is an ensemble of Classification Tree (CART) and a more efficient version of GB. This algorithm is very popular in the Machine Learning field, having its impact widely recognized in many machine learning and data mining challenges (Chen & Guestrin, 2016). Some of the modifications done to GB are parallel training (which fastens the algorithm), out-of-core computation (allowing that data is not loaded into memory), and sparse data optimization (in handling and speed up computation) (Bisong, 2019). Also, besides shrinkage and subsampling (used in GB) as regularization forms to control overfitting and attain better results, the XGBoost uses a regularized objective. More information about the XGBoost algorithm can be found at Chen & Guestrin (2016).

4.4.2. Test Design Generation

In order to choose the suitable model and model's hyperparameters, each constructed model should be tested and, therefore, a test design needs to be defined. The goal is that the obtained results while training/evaluating the model are close to the reality of how each model will perform. Therefore, both the dataset's characteristics and the way the model would be used were considered.

Firstly, regarding the data, a possible tendency in churn rate across the years has been observed for the different customer segments in the Data Understanding phase. Also, as mentioned before, the policy data maintains primarily stable through the years in most measures. However, significant changes usually happen in the policies' premium paid (and other variables it depends on). Moreover, given that each policy and its data only appear, at the most, once in each of the years, there is an annual (temporal) dependency on the data.

A possible approach to respect such temporal dependency is to use time-split cross-validation. Here, the model's generalization ability is assessed using the average of the performance metrics in the created data splits. At the same time, the temporal dependency is respected by always using the last block of data as validation. Given the applied increase in premium, the model should predict the probability of churn for the customers expected to renew each month in the year and therefore deliver monthly predictions. Given these model purposes, twelve splits were created (one split for each month in the year), using the validation set of the split's last available month, as shown in Figure 4.4.

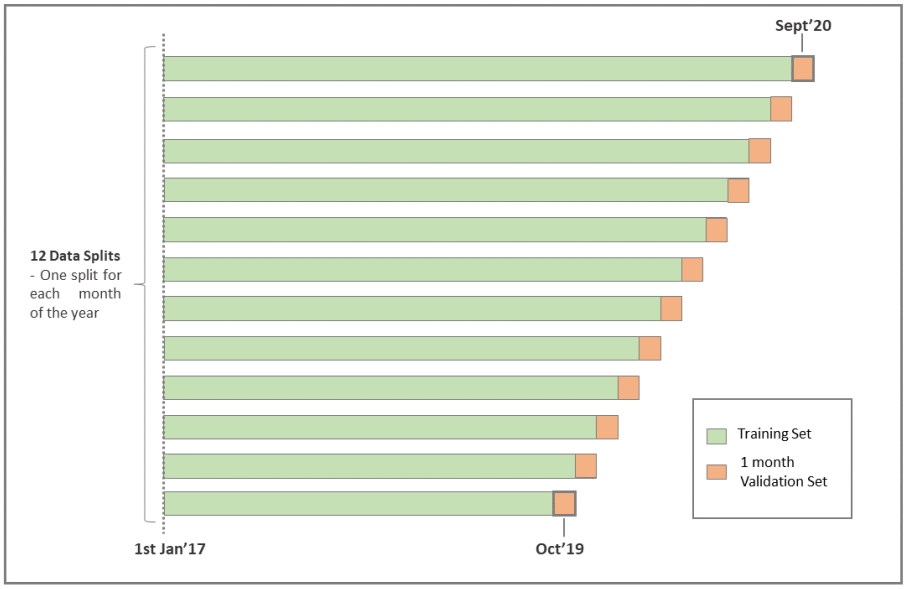


Figure 4.4 Representation of the designed time-split cross-validation

4.4.3. Model Construction and Assessment

4.4.3.1. Feature Selection with Ensemble Learning

As shown, one possible way to perform feature selection is to employ ensemble learning, and, for this, the approach needs to be designed. Given the popularity of heterogeneous feature selection methods, it was decided that this approach should be used, using the same subsets of data and features across the different learners. For this, many of the suggestions presented by Shah & Peretiatko (2021) were considered and are described below.

Firstly, as for the learner method level, many feature rankers were constructed using the PFI algorithm with a different base model to guarantee variability in the solutions. The selected models were the best set of base-line models (classifiers without parameter tunning) in terms of recall (true positive rate). Said that, the selected models were the ones that appeared to have a better understanding of the small class.

Then, as for combination level, it has been shown that several techniques are possible to combine the results and create a final rank, from simpler to more complex methods. Some of the more straightforward methods are using the median and mean of the results as aggregation rules. The median of the ranks was used for this work since it is less sensitive to outliers, being a more robust metric than the mean. However, features presenting the same median rank were afterward ordered by the mean of the observed ranks.

Lastly, for the threshold level decision, it has been shown that the most common approaches are considering a fixed threshold of the top-ranked features. However, such a technique comes with a few downsizes. In this way, once having each of the model's results, instead of selecting straightaway the top N features (given the difficulty of selecting the proper value of N), the used strategy was slightly different. After having the feature impact of each feature given by each of the learners, the total (sum) feature impact can be calculated (for each of the features summing all model's returned impacts), along with the cumulative impact (after ordering by the feature's total (sum) impact). Then, a given

ratio ($r \in]0, 1[$) of the total cumulative impact is applied and, the number of features that present a cumulative impact lower than the ratio of total cumulative impact will be the number of features to select (N) – N will be the cutting threshold. With this number (N), the Top N best-ranked features are selected, according to the ensemble's learners' opinions combination rule early defined. It is important to note that this technique will never select variables attributed with feature importance of zero or negative values.

However, using this approach, there is still a value that needs to be selected – the ratio (r). This ratio and its value decision will be explained further. As has been concluded in the past, the optimal feature size depends not only on the feature-label distribution but also on the used classifier (Hua et al., 2005). In this way, since each model is different, it was decided to create a more model-agnostic final ratio decision (individually for each of the selected models).

Additionally, an ordered feature list was created for each baseline model based on the PFI algorithm results using the given model only. In this case, for each of the classifiers is possible to choose the feature selection method that provides better results: using ensemble learning for feature selection or only the wrapper method - both having a common approach for selecting the number of features from the ranked list.

In the end, a recursive method was implemented to select the final feature list for each model. The total cumulative impact ratio (r) will decay until a specific stopping criterion is reached (the ratio starts at 1 and decays in each interaction). For each classifier, the best feature ranking method (comparing the ensemble and single model's ranked feature lists), data scaler (scaling only numerical variables), and ratio are selected considering the combination's performance in predicting the minority class (considering the recall using the designed cross-validation). In this way, the number of selected features depends on the dataset itself and the used classifier.

Feature List Creation

Given this, the used feature selection technique retrieves the best combination of:

- Feature Ranking Method – using the ensemble ranking or single model PFI's ranking
- Scaling – Scaling numerical features (with Standard Scaler or Robust Scaler) or without scaling
- Ratio – staring at 1 (using all features) and decaying 0.001 in each interaction; The ratio stops decreasing when three consecutive loops do not improve the cross-validated score (0.001 decay was decided based on the dataset's characteristics).

The best combination is the one with the best recall score in cross-validation. Also, if more than one combination presents the same recall, the one with fewer features is retrieved.

Feature List Optimization

After this, the created feature lists passed through the following approaches described below. Here, the final feature ranking is also used as an importance measure.

1. Try to add possible good features to each model's list
    - After the Data Understanding phase, for each of the customer's segments, a list of features is created with features that, based on this phase, are believed to be good predictors for the problem at hand.

- For each of the features that are not on the given created list, they are added (individually, one at the time) to the list to see if the recall score (in cross-validation) improves - if yes, should be added to the list (manually).

2. Try to remove possible bad/unnecessary features
- A primary stratified and shuffled train/test partition is created of 70/30 (to speed up given the numerous combinations of this phase)
- At each iteration, one of the features is left out (by inverse order of importance defined for the given features list in the later phase) - if the score improves/maintains, that specific feature is left out for the next interaction.
- After all the features are tested, the process repeats. The process will repeat until no changes are done - for the given iteration, no feature removal improves/maintains the recall score in the test split.
- Given the removed features for the primary train/test partition, each of the features removed is tested to be out in the model (individually, each feature at the time), and its cross-validated recall score is returned.
- If the average recall score in the validation sets improves/maintains, the feature is excluded (manually).

3. Try to replace numerical variables for its power transformation
- In the Data Preparation phase, Yeo-Johnson Power transformation was used to approximate the variable's distribution to normal
- Each iteration replaces one of the features by their transformation (by importance order/ranking). If the score improves, then its transformation for the next interaction replaces that specific feature.
- After all the features are tested, the process repeats. The process will repeat until no changes are done - no feature replacement improves the average recall score of validation sets.
- The selected transformed features are manually replaced in the final feature list.

4.4.3.2. Logistic Regression's Feature Selection with Backward Elimination

For the logistic regression's feature selection, an unregularized Logistic regression was constructed using all features (after the Data Preparation phase), and Backward Elimination as feature selection method was performed. As mentioned before, as the name suggests, the least significant features are iteratively removed until no other effect meets the specified level for removal (Bursac et al., 2008).

The removal criteria are based on the Wald test for individual parameters, using a significance level of 0.05 (the p-value cut-off was defined as 0.05). Here, the null hypothesis that the coefficient of the independent variable is equal to zero is tested versus an alternative hypothesis that the coefficient is nonzero, which can be written as: $H_0: \beta_x = 0 \; vs \; H_1: \beta_x \neq 0$ (Forthofer et al., 2007).

Besides the collected variables, some iterations and polynomial terms were introduced and tested for their significance in the model. Such interactions among variables were added whenever there were suspicions that the impact of one variable in the outcome also depended on another variable's value. Polynomial terms were added to cover/test cases where their relationship with the target is not linear.

Lastly, T-test was used to test possible (categorical) variable aggregations, testing if their model parameters were significantly different or not ($H_0: \beta_{x_i} - \beta_{x_j} = 0 \; vs \; H_1: \beta_{x_i} - \beta_{x_j} \neq 0$). For the cases in which the null hypotheses were not rejected, the aggregation was considered by summing both binary variables since, for the tested cases, the positive event is exclusive (never happens simultaneously in both).

Before this method was applied, features that presented low variances were removed to prevent multicollinearity. Such features can lead to a singular matrix (with determinant equaling to zero), which means that the matrix has no inverse, making it impossible to estimate the regression parameters.

The resulting set of features and corresponding summary statistics can be found in Appendix A.

4.4.3.3. Model Construction

Third-Party is the segment for which the highest percentage of total premium and the highest churn rate was observed. Therefore, the Third-Party segment was the first modeled segment and for which the results will be presented.

As said before, cross-validation was performed to assess the models and choose the correct parameters. Besides parameter tuning, in which multiple parameter combinations were tested to find the parameters that enhanced the results, other steps were taken in this phase. For each model constructed, the following described processes were proceeded.

Feature List and Data Scaler Choice

Here, the combination of feature list and data scaler (Standard Scaler, Robust Scaler, or no scaler) that guaranteed the best results for each model is selected and used in the subsequent phases.

Bellow, an example for Gradient Boosting model (without parameter tuning) for the Third-Party's dataset, comparing its cross-validated results using all dataset's features and the selected list, is present in Table 4.2. For this case, it is possible to observe that both results are similar, indicating that the lowest number of features should be considered only.

# Features	Specificity Train	Specificity Validation	AUC Train	AUC Validation	Recall Train	Recall Validation
84	**100%** +/- 0p.p.	**99.7%** +/- 0p.p.	**52.2%** +/- 0p.p.	**50.2%** +/- 1p.p.	**4.4%** +/- 1p.p.	**0.6%** +/- 1p.p.
37	**100%** +/- 0p.p.	**97.2%** +/- 9p.p.	**52.1%** +/- 0p.p.	**50.8%** +/- 2p.p.	**4.2%** +/- 1p.p.	**4.4%** +/- 13p.p.

Table 4.2 Gradient Boosting's feature list selection (without parameter tuning)

Data Pre-Processing using SMOTE

As has been mentioned, one of the possible approaches to deal with class imbalance is a data-level approach, where undersampling or oversampling can be used. Given the small dimensions of the datasets, using an undersampling technique was not appropriate and, therefore, an oversampling technique was attempted. Given SMOTE's popularity due to its simplicity and robustness, this technique generated new records into the training set.

Although it is shown that non-random sampling techniques can highly improve the classifier's performance, having a 1:1 distribution (the positive rate equaling the negative rate) might be unfavorable (Forman & Cohen, 2004). Therefore, smaller ratios of the minority class were attempted via a wrapper process so that the correct new ratio for the training sets' minority class for the model in question could be chosen.

Below, in Table 4.3, an example of the SMOTE's parameter choice that controls the new percentage of churn observed in the dataset is presented. It is important to note that this resampling is only done in the training set, so the reality in which the model will perform can be respected.

Churn Rate in Train	Specificity Train	Specificity Validation	AUC Train	AUC Validation	Recall Train	Recall Validation
original	**100%** +/- 0p.p.	**98.5%** +/- 2p.p.	**68.8%** +/- 3p.p.	**51.3%** +/- 2p.p.	**37.6%** +/- 5p.p.	**4.0%** +/- 5p.p.
50%	**98.9%** +/- 0p.p.	**23.4%** +/- 11p.p.	**95.9%** +/- 0p.p.	**54.4%** +/- 5p.p.	**92.8%** +/- 0p.p.	**85.3%** +/- 11p.p.
33%	**99.5%** +/- 3p.p.	**32.1%** +/- 29p.p.	**92.3%** +/- 0p.p.	**54.7%** +/- 0p.p.	**85.1%** +/- 1p.p.	**77.2%** +/- 18p.p.
23%	**99.8%** +/- 0p.p.	**43.5%** +/- 26p.p.	**87.2%** +/- 1p.p.	**54.8%** +/- 1p.p.	**74.5%** +/- 1p.p.	**66.1%** +/- 27p.p.
17%	**99.8%** +/- 0p.p.	**55.3%** +/- 31p.p.	**80.8%** +/- 1p.p.	**55.4%** +/- 4p.p.	**61.7%** +/- 2p.p.	**55.6%** +/- 30p.p.
13%	**100%** +/- 0p.p.	**63.3%** +/- 26p.p.	**75.6%** +/- 2p.p.	**54.9%** +/- 3p.p.	**51.2%** +/- 3p.p.	**46.5%** +/- 29p.p.
12%	**100%** +/- 0p.p.	**65.8%** +/- 32p.p.	**74.4%** +/- 1p.p.	**53.8%** +/- 4p.p.	**48.8%** +/- 2p.p.	**41.9%** +/- 32p.p.
11%	**100%** +/- 0p.p.	**67.1%** +/- 33p.p.	**73.3%** +/- 2p.p.	**53.0%** +/- 3p.p.	**46.6%** +/- 3p.p.	**38.9%** +/- 35p.p.

Table 4.3 Example for Extreme Gradient Boosting's - SMOTE's parameter choice (without parameter tuning)

Model Calibration & Threshold Tuning

As shown before, to obtain good models (in terms of prediction probabilities and probabilities' classification), the model's output probabilities should be calibrated, and the best decision threshold to classify the model's result into churn or renewal should be defined.

Regarding the probability calibration, the model predicted probabilities and the true values are used to assess each model's calibration performance using the calibration plot. Then, multiple copies of the model, using k-fold cross-validation, are fitted and, the probabilities predicted by these models are then calibrated on the hold-out sets. As calibration methods, Platt and Isotonic Calibration were tested. For this, the new predicted probabilities are assessed by plotting the calibration curve and comparing it with both, the perfect calibration line, and results before calibration.

For the choice of the threshold, the threshold with the optimal balance between false positive (specificity) and true positive rate (recall), this is the optimal threshold for the ROC curve, was located. With this knowledge, the metrics in cross-validation are optimized.

In order to accomplish this, the true positive rate or recall (TPR) and true negative rate or specificity (TNR) are computed for the predictions using a set of thresholds (this can also be used to create a ROC Curve plot). Then, the geometric mean (G-mean), which formula is presented below, represents the balance between both scores (the higher, the better).

$$GMean = \sqrt{(TPR * TNR)} \quad (6)$$

This metric is observed for the different attempted thresholds, and the threshold that maximizes this metric, this is, that optimizes the balance between TPR and TNR, is selected. Then, the optimal thresholds obtained across the different splits of the designed cross-validation are averaged into a final optimal threshold and are considered.

4.5. EVALUATION

In the Evaluation phase, the data mining results were assessed comparatively with the business success criteria. The process was also reviewed, and the final model was chosen for deployment.

The main goal of this project is to improve the profit margin for *Ocidental*'s Workers' Compensation branch by retaining customers that have intentions to leave given the premium variation suffered in the renovation process. A profit estimation analysis was performed to ensure that the final model would allow such an increase in the company's revenue.

For this, some suppositions were constructed based on the branch's business knowledge and are bellow explained. The results of this analysis are based on each model's confusion matrix results and other business metrics.

- **Revenue if the model predicts correctly that the customer renews (True Negatives - TN)**

$$Gain_{TN} = Client's\ Premium + \Delta\ Premium \quad (7)$$

  If the model predicts correctly that the customer will renew, the proposed premium variation will maintain, retaining both customer premium and premium variation.

- **Revenue if the model predicts that customer churns, but customer renews (False Positives - FP)**

$$Gain_{FP} = Client's\ Premium \quad (8)$$

  If the model predicts that the customer will churn, then there is a premium variation decrease (or even removal). Given that it is difficult to estimate the premium variation for those cases, the worst-case scenario (no increase) was considered.

- **Revenue if the model predicts correctly that the customer churns (True Positives - TP)**

$$Gain_{TP} = Client's\ Premium * retention\ ratio \quad (9)$$

The true positives are the customers that, if no model existed, would be lost. However, these customers can be preserved by correctly identifying their intentions and applying contingency measures (in this case, the decrease in the premium variation).
Nevertheless, these cases are similar to the False Positive (FP) cases, where the model predicts that these customers will also churn. Thus, there is still an increase in premium that is not accounted for in this estimation.
Besides this, it is also observable that it is not possible to preserve all risk by applying this contingency measure: customers still churn when there is no premium variation ($\Delta\ Premium = 0\%$). A possible explanation for such behavior is a better premium proposal in the competition that the company cannot match.
Therefore, it was decided to apply a retention ratio, considering that only $X\%$ of targeted churns can be reversed through the applied contingency measure. Given that, such value needed to be estimated.
It was observed that, by year, 13% of total churn in the Third-Party segment (with a standard deviation of 2 p.p.) happens for a 0% premium variation. Therefore, it was decided that such value should be rounded up considering its standard deviation and, as well, give a 100% margin as a safety measure. Thus, considering the retention ratio as:

$$retention\ ratio_{ThirdParty} = \big(1 - 2(0.13 + 0.02)\big) = 0.7 \quad (10)$$

- **Revenue if the model predicts that customer renews but customer Churns (False Negatives - FN)**

$$Gain_{FN} = 0 \quad (11)$$

Those are the cases where the model cannot foresee the customer's churn, so they are lost, having, or not having a model. Therefore, neither the customer's premium nor premium variation is retained.

Having estimated the different gains that each of the confusion matrix's measures, the increase in total revenue is given by the difference between the average revenue with the model implemented (To Be) and with no model (As Is):

$$Avg\ Increase\ Revene = Avg\ Revenue\ To\ Be - Avg\ Revenue\ As\ IS \quad (12)$$

where,

$$Avg\ Revenue\ As\ IS = \#Rwls * (Avg\ Client's\ Premium + Avg\ \Delta\ Premium) \quad (13)$$

and,

$$\begin{aligned} Avg\ Revenue\ ToBe = \ & \boldsymbol{TNR} * \#Rwls * Gain_{TN} + \boldsymbol{FPR} * \#Rwls * Gain_{FP} \\ & + \boldsymbol{TPR} * \#Chns * Gain_{TP} \end{aligned} \quad (14)$$

For each customer segment, the $\#Rwls$ is the number of verified renewals in a given year, the $\#Chns$ the number of verified churns in a given year, $\boldsymbol{TNR}$ the true negative rate and, $\boldsymbol{FPR}$ and $\boldsymbol{TPR}$ as false and true positive rates respectively.

Using the average metric's results ($\boldsymbol{TNR}, \boldsymbol{FPR}, \boldsymbol{TPR}$) obtained in each of the models and the observed values ($\#Rwls$, $\#Chs$, $Avg\ Client's\ Premium$, $Avg\ \Delta\ Premium$) in the last three years, it was possible to estimate, for each model, the expected average increase in revenue with their implementation.

$$Expected\ Avg\ Increase = \frac{1}{3} \sum_{i \in Y} (Avg\ Increase\ Revene_i) \qquad (15)$$

where Y are the last three available complete years.

4.6. DEPLOYMENT

Then, the model's deployment is planned, and its monitoring and maintenance plan is also designed. After a full review, the model will finally be integrated into the company's renewal pricing strategy and help the company increase the customer retention rate in the Workers' Compensation branch.

For this part, it was decided to deliver two different automated processes, described below. In both, Python is the primary tool used.

Monthly Predictions

The first deliverable aims to deliver monthly predictions for the month's potential renewals and is designed as follows.

In a particular month and year, the potential renewals data is collected using SAS Enterprise Guide. Python accesses the final output table and recreates in this dataset all of the necessary data preparation steps. At this stage of the process, the premium variation that each policy will suffer is unknown. The company wants to know each customer's reaction (probability to churn) to the different possible price increases. Therefore, many price increases scenarios are created, and each policy line will be recreated as many times as the number of different premiums increases.

The trained model is then used to predict the churn probability for each policy and different premium variation scenarios. Then, a final table, in which the different policies (rows), possible premium variation (columns), and customer reaction (probability of churn as value), is sent to SAS Enterprise Guide. Here, an optimization process (developed by the company) to choose the premium variation for each policy that maximizes the global profit will use the constructed table as input.

Model Performance Monitoring

The second deliverable aims to deliver a process in which the model's performance can be continuously assessed and further integrated into a dashboard.

This process is similar to the monthly predictions process described above. The only difference is that both the target (churn) and the premium variation are already known when it runs. The goal is to compare the model's past predictions with what happened (if policies have been churned or not) and understand its health. Therefore both, the data collection process (in SAS Enterprise Guide), data

preparation, and prediction (in Python) are made for the last five available months available (considering the current date with 3-month aging).

In the end, the model's output is compared with the observed target. With this, metrics are computed, and the profit estimation considering those metrics is returned to the user. In this way, the model can be monitored monthly, and if signs emerge that the model is outdated, the model should be repaired.

5. RESULTS AND DISCUSSION

5.1. RESULTS

Bellow, the results from the modeling phase will be presented. The different models constructed mentioned in this section can be found in Appendix B, along with the considered parameters, the used data scaling, number of features, percentage of the positive outcome in the dataset, and the probabilities' cutting threshold used.

5.1.1. Cross-Validation Results

In Table 5.1, the results of each model in the performed cross-validation (described in section 4.4) are presented. The results represent the average of each score in the 12 folds (comparing the train and validation set results) and the corresponding standard deviations.

Model	Specificity Train	Specificity Validation	AUC Train	AUC Validation	Recall Train	Recall Validation
GB	**74.4%** +/- 2p.p.	**72.3%** +/- 8p.p.	**62.7%** +/- 1p.p.	**60.1%** +/- 3p.p.	**50.9%** +/- 1p.p.	**47.9%** +/- 10p.p.
XGBoost	**74.8%** +/- 3p.p.	**72.5%** +/- 8p.p.	**61.7%** +/- 0p.p.	**59.2%** +/- 2p.p.	**48.7%** +/- 4p.p.	**46.0%** +/- 10p.p.
XGBoost with SMOTE	**96.0%** +/- 1p.p.	**81.0%** +/- 19p.p.	**63.7%** +/- 2p.p.	**54.9%** +/- 2p.p.	**31.4%** +/- 4p.p.	**28.8%** +/- 22p.p.
LR	**71.6%** +/- 1p.p.	**73.0%** +/- 4p.p.	**61.3%** +/- 0p.p.	**58.6%** +/- 4p.p.	**50.9%** +/- 1p.p.	**44.3%** +/- 9p.p.
MLP	**77.4%** +/- 1p.p.	**73.2%** +/- 9p.p.	**61.5%** +/- 1p.p.	**57.1%** +/- 4p.p.	**45.7%** +/- 2p.p.	**40.9%** +/- 14p.p.

Table 5.1 Cross-validation results

5.1.2. Model Assessment

In order to correctly assess each of the constructed models, these were tested using unseen data, the test set. As mentioned in section 3, the test set was extracted when all the phases mentioned above were concluded. Given this, and that 3-months of aging is required, it was possible to extract data for five months (from October 2020 to February 2021) and use it for testing. Bellow, in Table 5.2, the characteristics of each of the renewal months included in the test set are presented.

Test Set	Policies	Churn Rate
Oct '20	513	9.36%
Nov '20	458	6.55%
Dec '20	399	5.76%
Jan '21	1715	9.68%
Feb '21	523	8.99%

Table 5.2 Renewal months in the test set characteristics

Given the applied premium variation, each instance's churn probability was predicted using the available training data (until September 2020). Although performance results obtained in each set were observed individually for each of the unseen months present in the test set, the results were also averaged to evaluate each of the model's global performance.

The goal is to evaluate how the models behave at predicting customers' reactions for each renewal month. Therefore, the following tables present each model's performance in the test sets.

Test Set	Specificity	AUC	Recall
Oct '20	65.8%	60.0%	54.2%
Nov '20	68.0%	55.7%	43.3%
Dec '20	69.7%	54.4%	39.1%
Jan '21	91.2%	58.6%	25.9%
Feb '21	87.4%	62.8%	38.3%
Average	**76.4%**	**58.3%**	**40.2%**
+/- Standard Deviation	+/- 12p.p.	+/- 3p.p.	+/- 10p.p.

Table 5.3 Gradient Boosting (GB) performance in test sets

Test Set	Specificity	AUC	Recall
Oct '20	65.8%	61.0%	56.2%
Nov '20	67.3%	53.6%	40.0%
Dec '20	68.9%	54.0%	39.1%
Jan '21	90.0%	57.0%	24.1%
Feb '21	87.2%	60.6%	34.0%
Average	**75.8%**	**57.2%**	**38.7%**
+/- Standard Deviation	+/- 12p.p.	+/- 4p.p.	+/- 12p.p.

Table 5.4 Extreme Gradient Boosting (XGBoost) performance in test sets

Test Set	Specificity	AUC	Recall
Oct '20	87.7%	56.4%	25.0%
Nov '20	84.1%	52.1%	20.0%
Dec '20	77.9%	49.8%	21.7%
Jan '21	91.3%	53.8%	16.3%
Feb '21	87.0%	54.1%	21.3%
Average	**85.6%**	**53.2%**	**20.9%**
+/- Standard Deviation	+/- 5p.p.	+/- 5p.p.	+/- 3p.p.

Table 5.5 Extreme Gradient Boosting using SMOTE (XGBoost with SMOTE) performance in test sets

Test Set	Specificity	AUC	Recall
Oct ’20	71.2%	54.3%	37.5%
Nov ’20	69.9%	54.9%	40.0%
Dec ’20	75.0%	54.9%	34.8%
Jan ’21	84.6%	57.6%	30.7%
Feb ’21	77.5%	56.8%	36.2%
Average	**75.6%**	**55.7%**	**35.8%**
+/- Standard Deviation	+/- 6p.p.	+/- 1p.p.	+/- 4p.p.

Table 5.6 Logistic Regression (LR) performance in test sets

Test Set	Specificity	AUC	Recall
Oct ’20	70.8%	52.0%	33.3%
Nov ’20	73.4%	58.3%	43.3%
Dec ’20	71.5%	59.7%	47.8%
Jan ’21	92.2%	53.9%	15.7%
Feb ’21	88.2%	56.9%	25.5%
Average	**79.2%**	**56.2%**	**33.1%**
+/- Standard Deviation	+/- 10p.p.	+/- 3p.p.	+/- 13p.p.

Table 5.7 Multilayer Perceptron (MLP) performance in test sets

5.1.3. Model’s Evaluation

One of the business goals was to have an increase in profit. Bellow, in Figure 5.1, the estimated increase in revenue using the stated suppositions in section 4.5 is presented for each of the constructed models.

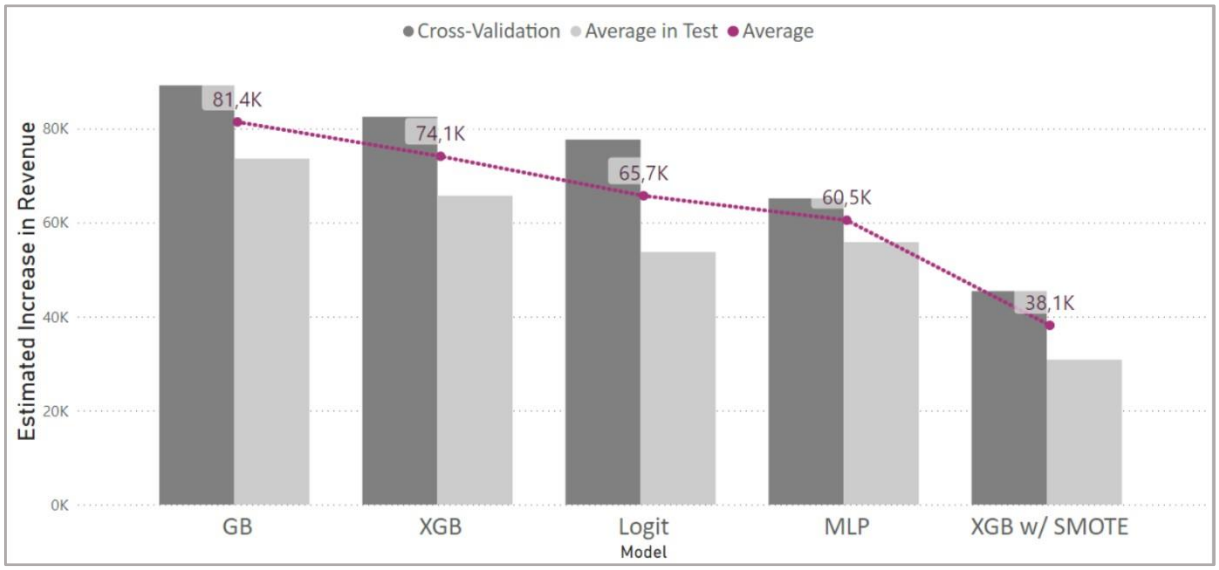


Figure 5.1 Estimated average increase in revenue considering the different model's results

Although all models assure an increase in revenue, the Gradient Boosting (GB) is the model for which the best results are expected, having the best global performance and the best induvial performance when considering the cross-validation and test results metrics. Therefore, and given that the business goal was successfully met, this was firstly considered the best model.

However, since the objective is to integrate the final model in the renewal's pricing process, as referred to in section 4.6, guaranteeing that the model allows a profit increase is not enough. The models need to allow the correct adjustment of the premium variation. Given this, for the same policy characteristics, the probability of churn should never decrease (or increase) when the premium variation increases (or decreases).

When such a relationship exists among a particular input and output variable (while all other variables maintain the same), it is called a monotonic relationship (Chuang et al., 2020). These relationships can be classified as monotonically increasing or decreasing, depending on whether both variables are positively or negatively correlated (respectively). Given this, the relationship between Premium Variation and Probability of Churn should be monotonically increasing since an increase in the Premium Variation should lead to an increase in the Probability of Churn.

The partial dependence plot allows having a global view of one feature effect on the model response, marginalizing over the values of all other input variables (Inouye et al., 2020). Therefore, by using partial dependence plots is possible to isolate other variable's impact and observe the Premium Variation's effect on the Probability of Churn. In Figure 5.2 it is possible to observe that this relationship is not monotonic for the Gradient Boosting model (best performing model).

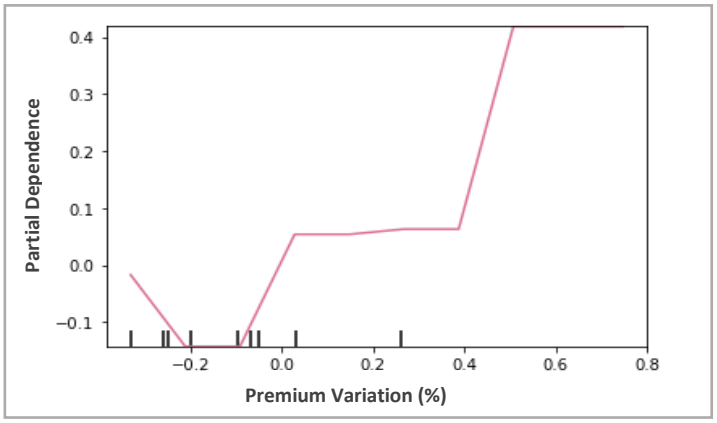


Figure 5.2 Gradient Boosting (GB) model expected target response as a function of Premium Variation (%)

Furthermore, the only model that is able to respect such relationship is the Logistic Regression, due to its form. However, adding an increasing monotonic constraint forcing such a relationship among both variables is possible for some models, particularly for the Extreme Gradient Boosting model from the *xgboost* package (Chen & Guestrin, 2016).

Therefore, a new Extreme Gradient Boosting model was constructed using the original dataset (without oversampling) as training set but, this time, guaranteeing that the Premium Variation feature bears an increasing monotonic relationship to the predicted response, by adding such constraint. Table 5.8 shows the obtained results in cross-validation, and Table 5.9 shows the obtained results for each

renewal month in the test set. Furthermore, the model's characteristics can be found in Appendix B and Appendix C.

Model	Specificity Train	Specificity Validation	AUC Train	AUC Validation	Recall Train	Recall Validation
Monotonic XGBoost	**75.0%** +/- 3p.p.	**74.4%** +/- 7p.p.	**59.9%** +/- 0p.p.	**59.0%** +/- 3p.p.	**44.9%** +/- 3p.p.	**43.5%** +/- 10p.p.

Table 5.8 Cross-validation results obtained with Extreme Gradient Boosting model imposing the described monotonic relationship (Monotonic XGBoost)

Test Set	Specificity	AUC	Recall
Oct '20	64.1%	59.1%	54.2%
Nov '20	64.0%	55.3%	46.7%
Dec '20	64.9%	54.2%	43.5%
Jan '21	87.3%	56.0%	24.7%
Feb '21	83.2%	57.6%	31.9%
Average **+/- Standard Deviation**	**72.7%** +/- 12p.p.	**56.4%** +/- 2p.p.	**40.2%** +/- 12p.p.

Table 5.9 Test sets results obtained with Extreme Gradient Boosting model imposing the described monotonic relationship (Monotonic XGBoost)

Although there is a slight decrease in performance, it is now possible to validate (Figure 5.3) that the relationship between Premium Variation and Probability of Churn now respect the deployment criteria, being monotonic increasing.

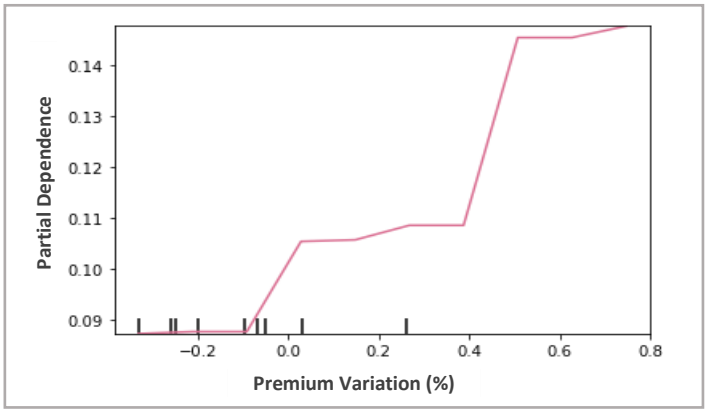


Figure 5.3 Expected target response as a function of Premium Variation (%) after applying monotonic constraint to the Extreme Gradient Boosting model

Moreover, in Figure 5.4 is possible to observe the estimated model's impact on revenue, compared with the only model that was able to assure such relationship among Premium Variation and the target.

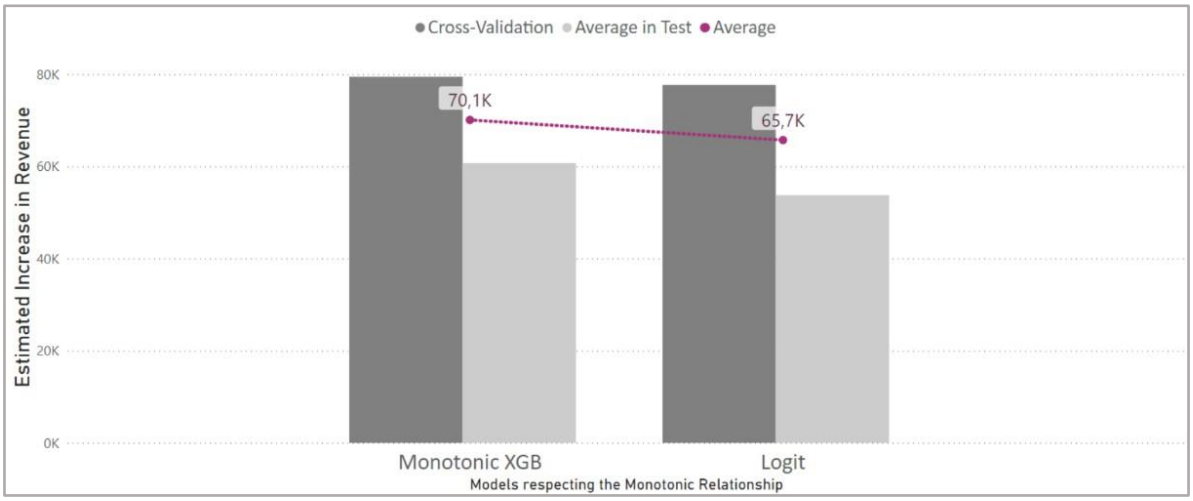


Figure 5.4 Estimated average increase in revenue for the Extreme Gradient Boosting model with the monotonic constraint and Logistic Regression

5.2. DISCUSSION

As stated earlier, the project had two business goals:

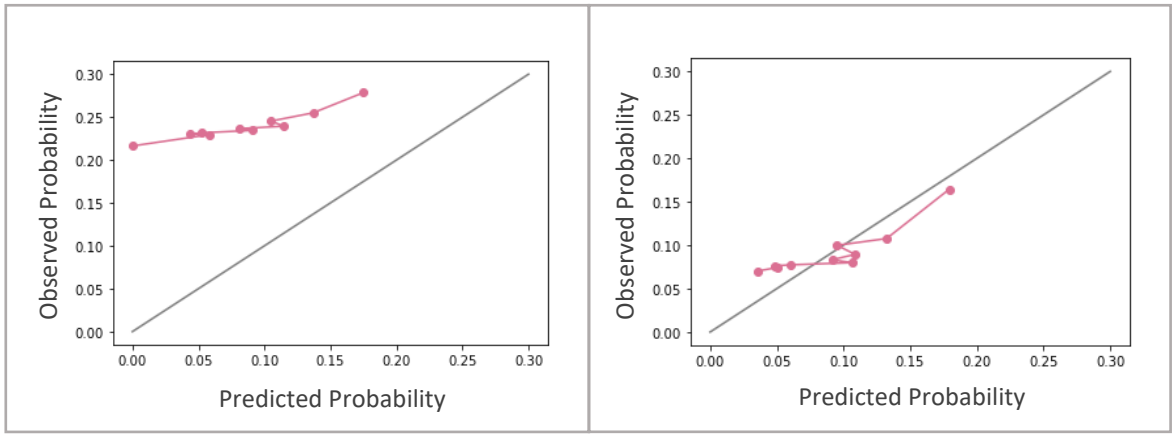
1. **Improve the branches' profit by correctly identifying customers who intended to churn (and renew) with the proposed premium variation.**
   It is crucial to correctly identify the customers that will renew with the applied variation as well. The company needs to maintain the premium variation (increase) applied to the policies in most of the cases since it significantly impacts profit.
2. **The selected model needs to be suitable for the renewal's pricing process.**
   The model aims to capture the customers' price elasticity. Therefore, a monotonic relationship between the target and Premium Variation needs to be assured.

In terms of predicting churn, there were already some signs of the complexity of the problem. Looking at the predictive variables' correlation with the target, the metric variable showing a higher correlation is Premium Variation (0.10 considering Pearson correlation and 0.08 of Spearman correlation), and the higher correlated non-metric variable is Flag Annual Payment (with a Crammer's V of 0.08). Moreover, the dataset's dimensions were small, and for the Third-Party case, only around 9% of instances were classified as churn. Such characteristics in the dataset can easily lead to problems such as overfitting. Therefore, it is essential to maintain the models simple.

Considering the model's construction phase, it was then possible to observe that models performed better, showing fewer signs of overfitting, when their complexity was reduced. For example, using fewer learners (for ensemble models) or fewer neurons/layers (for the neural networks). On the other hand, the reduction of the data-skewness by including synthetic samples (via SMOTE) in the training dataset did not improve results.

The results were also positively impacted by selecting a suitable prediction threshold (different than the default value of 0.5) allowing a suitable trade-off between the true positive (recall) and true negative rates (specificity). Moreover, given the deployed solution structure, the output probabilities appear to have benefited from calibration, approximating them to their true values. In Figure 5.5, it is

possible to observe the differences in the uncalibrated and calibrated predictions made by the Extreme Gradient Boosting model with the imposed monotonic relationship, using calibration plots. Observing Figure 5.5(a), is possible to observe that the uncalibrated model under-forecasts, this is the obtained probabilities are smaller than expected. Observing Figure 5.5(b), is possible to observe that calibrated probabilities are closer to diagonal line, therefore suggesting a better calibrated model.



(a) **(b)**

Figure 5.5 Calibration plots for XGBoost Monotonic predictions **(a)** Uncalibrated model Probabilities. **(b)** Calibrated model predictions.

By considering the obtained results, it is possible to observe the following:

- The constructed Extreme Gradient Boosting using SMOTE in data pre-processing has shown the highest level of overfitting and the lowest AUC on the validation and test sets.
- The highest AUC in both, cross-validation and average of test, is obtained for the Gradient Boosting (with an advantage of 0.9 p.p. in cross-validation and 1.1 p.p. in average of test results), followed by the Extreme Gradient Boosting model.

By evaluating the first business goal (profit increase), on average, the best results are obtained for the Gradient Boosting model. This model is also the leader when individually comparing the impact that the achieved metrics (in cross-validation and average of test sets) have on the estimated average increase in revenue.

However, considering the second business goal, only the Logistic Regression model could respect the monotonic relationship between the Premium Variation and the target. However, it was possible to force this monotonic relationship among both variables using an Extreme Gradient Boosting. By comparing this new model to the first business goal's leader (Gradient Boosting), it is possible to observe a decrease in the cross-validation's AUC (of 1.1 p.p.) and in the average of test's AUC (of 1.9 p.p.). Such reduction also impacts the estimated average increase in revenue, by around 11K €.

Though, this new model still has better results than the constructed Logistic Regression. It is only surpassed by the constructed (non-monotonic) Gradient Boosting and (non-monotonic) original Extreme Gradient Boosting.

6. CONCLUSIONS

This report results from a seven-month project developed in a one-year internship in *Grupo Ageas Portugal*. Bellow, the main conclusions of this project are presented.

Grupo Ageas Portugal is one of the largest insurance providers in Portugal, with different brands such as *Ocidental Seguros,* which provides offers in the life and non-life branches. This latter offers a Workers' Compensation insurance, which is mandatory in Portugal.

The *Ocidental*'s Workers' Compensation portfolio can be segmented into three main risk groups (Third-Party, Self-Employees, and Housekeepers), which have different renewal's pricing strategies and present different behaviors and available information.

Nowadays, customers' ease in exploring the available options allows them to change the insurance provider easily. Given the competitiveness of the insurance market, companies need to take action to retain their customers since it has a significant impact on their profit.

The renewal's pricing process occurs yearly at the customers' renewal date, proposing a variation to the policy's paid premium. The company's goal was to reduce the observed churn rates consequent of this process. Therefore, the company wanted to integrate a model that could identify early signs of churn accordingly to the different possible premium variations that the given policy could suffer. So, by capturing its customers' price elasticity, premium variations can be correctly adjusted, and customers that present a certain risk level of leaving, possibly maintained.

The CRISP-DM methodology was applied to accomplish this project, covering its main steps, and going back and forward when necessary.

The first part was dedicated to understanding the business itself, its goals, and which variables could provide helpful information about this customer behavior. Around 80 variables (numerical and categorical) were extracted from the company's database. However, as observed in the data understanding phase, not all variables were relevant, especially when separated within the three customer segments. This phase was instrumental in understanding some of the steps that should be done in the Data Preparation phase, where data in its raw form were prepared for modeling. Here, activities like redundant feature removal, handling missing values, creating new attributes, and formatting the data in a way that would allow (and help) the modeling phase were carried on. Moreover, as an alternative to using municipalities or districts, new geographical areas were created. For this, clustering analysis was performed and, for each of the customer segments, new geographical (contiguous) areas were created based on the municipalities' relevant demographic information and observed churn rates. Thus, it was possible to reduce eighteen new dimensions (in case districts were used) into only four relevant dimensions, using information regarding more than three hundred municipalities.

After having the dataset prepared, features to be used by each of the models are selected. The goal was to use a suitable number of features given the number of rows available for modeling. More features represent a higher problem complexity, and for which, models should have more data to understand. The features are chosen using ensemble learning, combining multiple learners' opinions about the feature's importance rankings. The Top N features are selected, for each model, based on

the results obtained for each set of features. For the Logistic regression, backward elimination based on the Wald's test was used.

The model's performance was evaluated across all the modeling phases. Time-split cross-validation was used so the existing temporal dependencies on the data could be respected. At the same time, the model's generalization ability can be guaranteed, approximating the validation results to the actual model performance.

The final model selection decision was based on the company's two business goals: increasing the branches' profit (accordingly with the model's performance) and have a suitable model for optimizing the renewal's pricing process. For this latter, a monotonic increasing relationship among Premium Variation and Probability of Churn needs to be assured.

The final selected model was an Ensemble Learning technique, the Extreme Gradient Boosting model for which the monotonic relationship was forced. This model shows in cross-validation an AUC of 59.0% (with a standard deviation of 3 p.p.), a specificity of 74.4% (with a standard deviation of 7 p.p.) and, lastly, a recall of 43.5% (with a standard deviation of 10 p.p.). As for the average results in the used test sets, an AUC of 56.4% (with a standard deviation of 2 p.p.), a specificity of 72.7% (with a standard deviation of 12 p.p.) and, lastly, a recall of 40.2% (with a standard deviation of 12 p.p.) were observed.

Furthermore, this model can answer the company's needs. Firstly, it is suitable for the optimization of the renewal's pricing process: for a given renewal year and policy, the delivered model allows that if the company decides to increase (or decrease) the premium variation, then the probability of churn will either maintain the same value or increase (or decrease) in its value. Therefore, permitting data-driven decisions. Lastly, the model allows an increase in profit: it was estimated that this model could contribute to an annual average increase of around 70K €.

7. LIMITATIONS AND RECOMMENDATIONS FOR FUTURE WORKS

It is possible to notice that the delivered solution has some space for improvement since the results in terms of performance measures could be higher. Results are highly dependent on the data used and, here, some improvements could be made, or other techniques attempted, like for example the test of other classification algorithms.

Several issues were identified during the data collection task. Many inconsistencies were detected among the different data sources (the company is currently constructing a unified source of information). Although such problems were dealt within the best way possible, different quality issues have emerged, impacting the final collected dataset. Therefore, the quality in the used dataset for modeling could not be assured. It was also not possible to collect data for many of the company's policies, impacting the final dataset's dimensions. Even having data for four whole years, a higher amount of data could surely help models in their results. Additionally, there are possibly other variables that are not available to the company but could be relevant, like external factors that might influence the customer's behavior.

Regarding data pre-processing, other techniques could be tested as other missing values imputation techniques, a deeper outlier detection analysis, other oversampling techniques, and further dimensionality reduction (feature selection) techniques could be explored as well.

Lastly, as was possible to observe, some models present better results than others. Therefore, it could be interesting to explore other models that are monotonic or allow a monotonic constraint. Two models that would be interesting to test are LightGBM (from Python's *LightGBM* package (Ke et al. 2017)) and lattice-based models (from Python's *TensorFlow-Lattice* package (Google AI Blog, 2017)). Similarly to the Extreme Gradient Boosting model (from the *xgboost* package (Chen & Guestrin, 2016)), these models allow the introduction of monotonic constraints. However, due to security constraints that the company needs to ensure, it was not possible to obtain such packages in a viable time to this project execution.

8. BIBLIOGRAPHY

Ahn, J., Hwang, J., Kim, D., Choi, H., & Kang, S. (2020). A Survey on Churn Analysis in Various Business Domains. *IEEE Access*, *8*, 220816–220839. https://doi.org/10.1109/ACCESS.2020.3042657

Ayer, M., Brunk, H. D., Ewing, G. M., Reid, W. T., & Silverman, E. (1955). An empirical distribution function for sampling with incomplete information. *The Annals of Mathematical Statistics*, *26*(4), 641–647. https://doi.org/10.1214/aoms/1177728423

Batista, G. E., & Monard, M. C. (2003). An analysis of four missing data treatment methods for supervised learning. *Applied Artificial Intelligence*, *17*(5–6), 519–533. https://doi.org/10.1080/713827181

Bellman, R. (1966). Dynamic programming. *Science*, *153*(3731), 34–37. https://doi.org/10.1126/science.153.3731.34

Bisong, E. (2019). Building Machine Learning and Deep Learning Models on Google Cloud Platform. In *Apress, Berkeley, CA*. https://doi.org/10.1007/978-1-4842-4470-8_29

Bolancé, C., Guillen, M., & Padilla-Barreto, A. E. (2016). Predicting Probability of Customer Churn in Insurance. In R. León, M. Muñoz-Torres, & J. Moneva (Eds.), *Modeling and Simulation in Engineering, Economics and Management. MS 2016. Lecture Notes in Business Information Processing, vol 254* (pp. 82–91). Springer, Cham. https://doi.org/10.1007/978-3-319-40506-3_9

Bollegala, D. (2017). Dynamic feature scaling for online learning of binary classifiers. *Knowledge-Based Systems*, *129*, 97–105. https://doi.org/10.1016/j.knosys.2017.05.010

Bolón-Canedo, V., & Alonso-Betanzos, A. (2019). Ensembles for feature selection: A review and future trends. *Information Fusion*, *52*, 1–12. https://doi.org/10.1016/j.inffus.2018.11.008

Box, G. E., & Cox, D. R. (1964). An Analysis of Transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, *26*(2), 211–243. https://doi.org/10.1111/j.2517-6161.1964.tb00553.x

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1023/A:1010933404324

Bursac, Z., Gauss, C. H., Williams, D. K., & Hosmer, D. W. (2008). Purposeful selection of variables in logistic regression. *Source Code for Biology and Medicine*, *3*(17). https://doi.org/10.1186/1751-0473-3-17

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*(1), 321–357. https://doi.org/10.5555/1622407.1622416

Chawla, N. V., Cieslak, D. A., Hall, L. O., & Joshi, A. (2008). Automatically countering imbalance and its empirical relationship to cost. Data Mining and Knowledge Discovery, 17, 225–252. https://doi.org/10.1007/s10618-008-0087-0

Chawla, N. V., Japkowicz, N., & Kotcz, A. (2004). Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explor., 6*, 1-6. https://doi.org/10.1145/1007730.1007733

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chiew, K. L., Tan, C. L., Wong, K., Yong, K. S., & Tiong, W. K. (2019). A new hybrid ensemble feature selection framework for machine learning-based phishing detection system. *Information Sciences*, *484*, 153–166. https://doi.org/10.1016/j.ins.2019.01.064

Chuang, H. C., Chen, C. C., & Li, S. T. (2020). Incorporating monotonic domain knowledge in support vector learning for data mining regression problems. *Neural Computing and Applications*, *32(15)*, 11791–11805. https://doi.org/10.1007/s00521-019-04661-4

Collell, G., Prelec, D., & Patil, K. R. (2018). A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. *Neurocomputing*, *275*, 330–340. https://doi.org/10.1016/j.neucom.2017.08.035

De Winter, J. C., Gosling, S. D., & Potter, J. (2016). Supplemental Material for Comparing the Pearson and Spearman Correlation Coefficients Across Distributions and Sample Sizes: A Tutorial Using Simulations and Empirical Data. *Psychological Methods*, *21*(3), 273–290. https://doi.org/10.1037/met0000079.supp

Dietterich, T. G. (2002). Ensemble Learning. *The Handbook of Brain Theory and Neural Networks*, *2*(1), 110–125.

Dolatabadi, S. H., & Keynia, F. (2017). Designing of Customer and Employee Churn Prediction Model Based on Data Mining Method and Neural Predictor. *2017 2nd International Conference on Computer and Communication Systems (ICCCS)*, 74–77. https://doi.org/10.1109/CCOMS.2017.8075270

Dormann, C. F. (2020). Calibration of probability predictions from machine-learning and statistical models. *Global Ecology and Biogeography*, *29*(4), 760–765. https://doi.org/10.1111/geb.13070

El Aboudi, N., & Benhlima, L. (2016). Review on Wrapper Feature Selection Approaches. *2016 International Conference on Engineering & MIS (ICEMIS)*, 1–5. https://doi.org/0.1109/ICEMIS.2016.7745366

El Bouchefry, K., & De Souza, R. S. (2020). Learning in Big Data: Introduction to Machine Learning. In *Knowledge Discovery in Big Data from Astronomy and Earth Observation* (pp. 225–249). Elsevier. https://doi.org/10.1016/b978-0-12-819154-5.00023-0

Fernández, A., García, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for Learning from Imbalanced Data: Progress and Challenges, Marking the 15-year Anniversary. *Journal of Artificial Intelligence Research*, *61*, 863–905. https://doi.org/10.1613/jair.1.11192

Forman, G., & Cohen, I. (2004). Learning from little: Comparison of classifiers given little training. In J. Boulicaut, F. Esposito, F. Giannotti, & D. Pedreschi (Eds.), *Knowledge Discovery in Databases: PKDD 2004. PKDD 2004. Lecture Notes in Computer Science, vol 320* (pp. 161–172). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-30116-5_17

Forthofer, R. N., Lee, E. S., & Hernandez, M. (2007). Logistic and Proportional Hazards Regression. In *Biostatistics (Second Edition)* (pp. 387–419). Academic Press. https://doi.org/10.1016/B978-0-12-369492-8.50019-4

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2

García, S., Luengo, J., & Herrera, F. (2016). Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. *Knowledge-Based Systems*, *98*, 1–29. https://doi.org/10.1016/j.knosys.2015.12.006

Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences. *Atmospheric Environment*, *32*(14–15), 2627–2636. https://doi.org/10.1016/S1352-2310(97)00447-0

Günther, C. C., Tvete, I. F., Aas, K., Sandnes, G. I., & Borgan, Ø. (2014). Modelling and predicting customer churn from an insurance company. *Scandinavian Actuarial Journal*, *2014*(1), 58–71. https://doi.org/10.1080/03461238.2011.636502

Google AI Blog (2017). Tensorflow lattice: Flexibility empowered by prior knowledge. Retrived from https://ai.googleblog.com/2017/10/tensorflow-lattice-flexibility.html

Harris, R., Sleight, P., & Webber, R. (2005). Introducing Geodemographics. In *Geodemographics, GIS and neighbourhood targeting* (pp. 16–17). John Wiley & Sons.

He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

He, Y., Xiong, Y., & Tsai, Y. (2020). Machine Learning Based Approaches to Predict Customer Churn for an Insurance Company. *2020 Systems and Information Engineering Design Symposium (SIEDS)*, 1–6. https://doi.org/10.1109/SIEDS49339.2020.9106691

Hua, J., Xiong, Z., Lowey, J., Suh, E., & Dougherty, E. R. (2005). Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, *21*(8), 1509–1515. https://doi.org/10.1093/bioinformatics/bti171

Inouye, D. I., Leqi, L., Kim, J. S., Aragam, B., & Ravikumar, P. (2020). Automated Dependence Plots. *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, *124*, 1238–1247. http://arxiv.org/abs/1912.01108

Jain, A. K., & Chandrasekaran, B. (1982). Dimensionality and sample size considerations in pattern recognition practice. In *Handbook of Statistics* (Vol. 2, pp. 835–855). https://doi.org/10.1016/S0169-7161(82)02042-2

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, 18(1), 559-563.

Kalousis, A., Prados, J., & Hilario, M. (2007). Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and Information Systems*, *12*, 95–116. https://doi.org/10.1007/s10115-006-0040-8

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W. Ma, W., Ye, Q., Liu, T.Y (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems, 30,* 3146-3154.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1–2), 273–324. https://doi.org/10.1016/S0004-3702(97)00043-X

Kumar, V., & Minz, S. (2014). Feature Selection: A literature Review. *The Smart Computing Review*, *4*(3), 211–229. https://doi.org/10.6029/smartcr.2014.03.007

Maletic, J., & Marcus, A. (2000). Data Cleansing: Beyond Integrity Analysis. *Iq*, 200–209.

McCue, C. (2015). Identification, Characterization, and Modeling. In *Data Mining and Predictive Analysis (Second Edition)* (pp. 137–155). https://doi.org/10.1016/B978-0-12-800229-2.00007-9

McGovern, A., Lagerquist, R., John Gagne, D., Jergensen, G. E., Elmore, K. L., Homeyer, C. R., & Smith, T. (2019). Making the Black Box More Transparent: Understanding the Physical Implications of Machine Learning. *Bulletin of the American Meteorological Society*, *100*(11), 2175–2199. https://doi.org/10.1175/BAMS-D-18-0195.1

Meir, R., & Rätsch, G. (2003). An Introduction to Boosting and Leveraging. In S. Mendelson & A. J. Smola (Eds.), *Advanced lectures on machine learning* (pp. 118–183). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-36434-X_4

Naeini, M. P., Cooper, G. F., & Hauskrecht, M. (2015). Obtaining Well Calibrated Probabilities Using Bayesian Binning. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2901–2907.

Natekin, A., & Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics*, *7*, 21. https://doi.org/10.3389/fnbot.2013.00021

Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. *Proceedings of the 22nd International Conference on Machine Learning*, 625–632. https://doi.org/10.1145/1102351.1102430

Njikam, A. N. S., & Zhao, H. (2016). A novel activation function for multilayer feed-forward neural networks. *Applied Intelligence*, *45*, 75–82. https://doi.org/10.1007/s10489-015-0744-0

Osborne, J. W. (2010). Improving your data transformations: Applying the Box-Cox transformation. *Practical Assessment, Research and Evaluation*, *15*(12). https://doi.org/10.7275/qbpc-gk17

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12,* 2825–2830. Retreived from https://jmlr.csail.mit.edu/papers/volume12/pedregosa11a/pedregosa11a.pdf

Pham, D. T., Dimov, S. S., & Nguyen, C. D. (2005). Selection of K in K-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *219*(1), 103–119. https://doi.org/10.1243/095440605X8298

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, *6*(3), 21–45. https://doi.org/10.1109/MCAS.2006.1688199

Pozzolo, A. D., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating Probability with Undersampling for Unbalanced Classification. *2015 IEEE Symposium Series on Computational Intelligence*, 159–166. https://doi.org/10.1109/SSCI.2015.33

Price, B. (2002). Making CRM come to life. *E-Business Review*, 25–31.

Provost, F. (2008). *Machine learning from imbalanced data sets 101*.

Rokach, L., & Maimon, O. (2005). Clustering Methods. In O. Maimon & L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook* (pp. 321–352). Springer, Boston, MA. https://doi.org/10.1007/0-387-25465-X_15

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*(3), 581–592. https://doi.org/10.1093/biomet/63.3.581

Saeys, Y., Abeel, T., & Van De Peer, Y. (2008). Robust Feature Selection Using Ensemble Feature Selection Techniques. In W. Daelemans (Ed.), *Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2008. Lecture Notes in Computer Science, vol 5212* (pp. 313–325). Springer-Verlag Berlin Heidelberg 2008. https://doi.org/10.1007/978-3-540-87481-2_21

Sasser, W. E., & Reichheld, F. F. (1990). Zero Defections -Quality Comes to Services. *Harvard Business Review*, *68*(5), 105–111.

Scriney, M., Nie, D., & Roantree, M. (2020). Predicting Customer Churn for Insurance Data. In M. Song, I. Song, G. Kotsis, A. M. Tjoa, & I. Khali (Eds.), *Big Data Analytics and Knowledge Discovery. DaWaK 2020. Lecture Notes in Computer Science, vol 12393* (pp. 256–265). Springer, Cham. https://doi.org/10.1007/978-3-030-59065-9_21

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference.* (Vol. 57, pp. 61).

Seijo-Pardo, B., Bolón-Canedo, V., & Alonso-Betanzos, A. (2017). Testing Different Ensemble Configurations for Feature Selection. *Neural Processing Letters*, *46*, 857–880. https://doi.org/10.1007/s11063-017-9619-1

Seijo-Pardo, B., Bolón-Canedo, V., Porto-Díaz, I., & Alonso-Betanzos, A. (2015). Ensemble Feature Selection for Rankings of Features. In I. Rojas, G. Joya, & A. Catala (Eds.), *Advances in Computational Intelligence. IWANN 2015. Lecture Notes in Computer Science, vol 9095*. Springer, Cham. https://doi.org/10.1007/978-3-319-19222-2_3

Shah, R., & Peretiatko, V. (2021). Using Feature Importance Rank Ensembling (FIRE) for Advanced Feature Selection. Retrived from https://www.datarobot.com/blog/using-feature-importance-rank-ensembling-fire-for-advanced-feature-selection/

Spiteri, M., & Azzopardi, G. (2018). Customer Churn Prediction for a Motor Insurance Company. *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, 173–178. https://doi.org/10.1109/ICDIM.2018.8847066

Strobl, C., Boulesteix, A. L., Kneib, T., Augustin, T., & Zeileis, A. (2008). Conditional variable importance for random forests. *BMC Bioinformatics*, *9*, 307. https://doi.org/10.1186/1471-2105-9-307

Sundarkumar, G. G., & Ravi, V. (2015). A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance. *Engineering Applications of Artificial Intelligence*, *37*, 368–377. https://doi.org/10.1016/j.engappai.2014.09.019

Talabis, M., McPherson, R., Miyamoto, I., & Martin, J. (2015). Analytics Defined. In *Information security analytics: finding security insights, patterns, and anomalies in big data* (pp. 1–12). Syngress. https://doi.org/10.1016/b978-0-12-800207-0.00001-0

Toloşi, L., & Lengauer, T. (2011). Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics*, *27*(14), 1986–1994. https://doi.org/10.1093/bioinformatics/btr300

Vafeiadis, T., Diamantaras, K. I., Sarigiannidis, G., & Chatzisavvas, K. C. (2015). A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*, *55*, 1–9. https://doi.org/10.1016/j.simpat.2015.03.003

Verleysen, M., & François, D. (2005). The Curse of Dimensionality in Data Mining and Time Series Prediction. In J. Cabestany, A. Prieto, & ｌF. Sandova (Eds.), *Computational Intelligence and Bioinspired Systems. IWANN 2005. Lecture Notes in Computer Science, vol 3512* (pp. 758–770). Springer, Berlin, Heidelberg. https://doi.org/10.1007/11494669_93

Wang, G., Hao, J., Ma, J., & Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications*, *38*(1), 223–230. https://doi.org/10.1016/j.eswa.2010.06.048

Wang, L., Zhang, Z., Zhang, X., Zhou, X., Wang, P., & Zheng, Y. (2021). A Deep-forest based approach for detecting fraudulent online transaction. *Advances in Computers*, *120*, 1–38. https://doi.org/10.1016/bs.adcom.2020.10.001

Wang, S. & Yao, X. (2012). Multiclass Imbalance Problems: Analysis and Potential Solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *42*(4), 1119–1130. https://doi.org/10.1109/TSMCB.2012.2187280

Wirth, R., & Hipp, J. (2000). CRISP-DM : Towards a Standard Process Model for Data Mining. *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining (Vol. 1)*.

Xu, R., & Wunsch, D. (2005). Survey of Clustering Algorithms. *IEEE Transactions on Neural Networks*, *16*(3), 645–678. https://doi.org/10.1109/TNN.2005.845141

Yap, B. W., Rani, K. A., Rahman, H. A., Fong, S., Khairudin, Z., & Abdullah, N. N. (2014). An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets. In T. Herawan, M. Deris, & J. Abawajy (Eds.), *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013). Lecture Notes in Electrical Engineering, vol 285*. Springer, Singapore. https://doi.org/10.1007/978-981-4585-18-7_2

Yeo, I. K. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, *87*(4), 954–959. https://doi.org/10.1093/biomet/87.4.954

Zell, A. (1994). *Simulation Neuronaler Netze* (Vol. 1, Issue 5.3). Addison-Wesley Bonn.

Zhang, T., & Yang, B. (2017). Box–Cox Transformation in Big Data. *Technometrics*, *59*(2), 189–201. https://doi.org/10.1080/00401706.2016.1156025

Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, *58*(Part B), 308–324. https://doi.org/10.1016/j.trc.2015.02.019

Zhao, X., Wang, Y., & Cha, H. W. (2009). A Mathematics Model of Customer Churn Based on PCA Analysis. *2009 International Conference on Computational Intelligence and Software Engineering*, 1–5. https://doi.org/10.1109/CISE.2009.5365094

Zhou, J., Li, E., Wang, M., Chen, X., Shi, X., & Jiang, L. (2019). Feasibility of Stochastic Gradient Boosting Approach for Evaluating Seismic Liquefaction Potential Based on SPT and CPT Case Histories. *Journal of Performance of Constructed Facilities*, *33*(3), 04019024. https://doi.org/10.1061/(ASCE)CF.1943-5509.0001292

9. APPENDIX

9.1. APPENDIX A - BACKWARD ELIMINATION (LOGISTIC REGRESSION FINAL FEATURE LIST)

Bellow in Figure 9.1, the final summary statistics after performing Backward Elimination, for selecting the more significant variables using *statsmodels* package (Seabold & Perktold, 2010) in Python, is presented. This process produced the final used feature list for the regularized logistic regression (from *scikit-learn* package (Pedregosa et al., 2011)).

Optimization terminated successfully.
Current function value: 0.289935
Iterations 7

Logit Regression Results

Dep. Variable:	Churn	No. Observations:	22637
Model:	Logit	Df Residuals:	22610
Method:	MLE	Df Model:	26
Date:	Sat, 17 Jul 2021	Pseudo R-squ.:	0.05145
Time:	13:52:14	Log-Likelihood:	-6563.3
converged:	True	LL-Null:	-6919.3
Covariance Type:	nonrobust	LLR p-value:	2.193e-133

	coef	std err	z	P>|z|	[0.025	0.975]
const	-2.3516	0.153	-15.384	0.000	-2.651	-2.052
V2.(Category 8)	-0.2247	0.038	-5.975	0.000	-0.298	-0.151
V8.(Category 9)	-0.1615	0.052	-3.108	0.002	-0.263	-0.060
V4.(Category 1)	0.3039	0.044	6.958	0.000	0.218	0.390
Premium Variation(%)	2.7126	0.306	8.876	0.000	2.114	3.312
V3.(Category 1)	4.332e-05	1.67e-05	2.595	0.009	1.06e-05	7.6e-05
V5.(Category 8) *V9.(Category 8)	-6.992e-06	2e-06	-3.489	0.000	-1.09e-05	-3.06e-06
V10.(Category 4)	-0.1591	0.074	-2.164	0.030	-0.303	-0.015
North Coast Cluster (Flag)	0.1664	0.050	3.335	0.001	0.069	0.264
V11.(Category 4)	-0.2378	0.063	-3.747	0.000	-0.362	-0.113
V12.(Category 4)	0.5996	0.261	2.301	0.021	0.089	1.110
V1.(Category 1)	0.8355	0.055	15.133	0.000	0.727	0.944
V13.(Category 4)	0.3555	0.078	4.568	0.000	0.203	0.508
V14.(Category 7)	0.3123	0.086	3.621	0.000	0.143	0.481
V15.(Category 7)	0.1495	0.062	2.395	0.017	0.027	0.272
V16.(Category 4)	-0.6964	0.226	-3.076	0.002	-1.140	-0.253
V17.(Category 4)	-0.2442	0.099	-2.478	0.013	-0.437	-0.051
V18.(Category 4)	-0.4572	0.130	-3.520	0.000	-0.712	-0.203
V19.(Category 4)	-0.3288	0.157	-2.100	0.036	-0.636	-0.022
V20.(Category 1)	-0.2558	0.090	-2.845	0.004	-0.432	-0.080
V21.(Category 4)	-0.3707	0.145	-2.557	0.011	-0.655	-0.087
Premium Variation (%) ^2	-0.8600	0.145	-5.924	0.000	-1.145	-0.575
Premium Variation (%) ^3	0.0552	0.013	4.328	0.000	0.030	0.080
V3.(Category 1)* Premium Variation (%)	0.0002	5.74e-05	3.262	0.001	7.47e-05	0.000
V2.(Category 8) ^2	0.0180	0.005	3.473	0.001	0.008	0.028
V2.(Category 8) ^3	-0.0005	0.000	-2.721	0.006	-0.001	-0.000
V4.(Category 8) ^2	-0.0148	0.004	-3.306	0.001	-0.024	-0.006

Figure 9.1 Final summary statistics after performing backward elimination

9.2. APPENDIX B - CONSTRUCTED MODELS' CHARACTERISTICS

Model	Algorithm	Package (Python)	Parameters	# Features	Scaling Method	Churn Rate (* if SMOTE)	Thres-hold
GB	Gradient Boosting	*scikit-learn* (Pedregosa et al., 2011)	**Loss Function:** Deviance; **Shrinkage:** 0.02; **Estimators:** 80; **Subsample (Column and Row-level):** 100%; **Split Quality:** Friedman's MSE; **Min Samples Split:** 2; **Min Samples Leaf:** 1; **Max Depth:** 3; **Min Impurity Decrease:** 0	37	Standard Scaler	9.1%	0.10
XGBoost	Extreme Gradient Boosting	*xgboost* (Chen & Guestrin, 2016)	**Shrinkage:** 0.02; **Estimators:** 80; **Subsample (Column/Row-level):** 80%/80%; **Min Loss Reduction:** 0.3; **Maximum Delta Step:** 5; **Max Depth:** 3; **L1 regularization:** 0; **L2 regularization:** 1; **Min Samples' Weight Leaf:** 0.5; **Learning Objective:** Logistic	36	Standard Scaler	9.1%	0.096
XGBoost with SMOTE	Extreme Gradient Boosting	*xgboost* & *SMOTE* (Chen & Guestrin, 2016; Lemaître et al., 2017)	**Shrinkage:** 0.08; **Estimators:** 100; **Subsample (Column/Row-level):** 80%/80%; **Min Loss Reduction:** 0.3; **Max Depth:** 4; **L1 regularization:** 0; **L2 regularization:** 0.9; **Min Samples' Weight Leaf:** 1; **Learning Objective:** Logistic; **Data Balance Control (scale_pos_weight):** 1.3	37	Standard Scaler	13%*	0.155
LR	Logistic Regression	*scikit-learn* (Pedregosa et al., 2011)	**Inverse Regularization Strength:** 0.1; **Regularization Type:** L2 (squared weights); **Solver:** L-BFGS; **Tolerance for Sopping Criteria (Convergence):** 10^{-4}; **Constant (Intercept):** True;	26	Standard Scaler	9.1%	0.105
MLP	Multilayer Perceptron	*scikit-learn* (Pedregosa et al., 2011)	**Hidden Layer Sizes:** (54,16,); **Number of Epochs:** 15; **Batch Size:** 200; **Solver:** Adam; **Learning Rate Initialization:** 0.001; **Decay Rate (1st Moment estimate)**: 0.9; **Decay Rate (2nd Moment estimate):** 0.999; **Smoothing parameter**: 10^{-8}; **Activation Function:** ReLU; **Batch Sizes**: 200;	31	Standard Scaler	9.1%	0.095
Monotonic XGBoost	Extreme Gradient Boosting	*xgboost* (Chen & Guestrin, 2016)	**Shrinkage:** 0.02; **Estimators:** 50; **Subsample (Column/Row-level):** 80%/80%; **Min Loss Reduction:** 0.3; **Maximum Delta Step:** 5; **Max Depth:** 2; **L1 regularization:** 0; **L2 regularization:** 1; **Min Samples' Weight Leaf:** 0.5; **Learning Objective:** Logistic; **Monotone Constraints:** Premium Variation (%)	27	Standard Scaler	9.1%	0.096

Table 9.1 Different constructed models' characteristics

9.3. APPENDIX C - FINAL MODEL'S FEATURE LIST

The final used feature list was selected using the constructed Feature importance ranked ensemble technique (described in section 4.4).

Bellow in Figure 9.2 is shown the features' importance using the "gain" importance type, plotted using the *xgboost* package (Chen & Guestrin, 2016). Here, it is possible to see how each feature contributed to the final Monotonic XGBoost model based on the average gain of splits that use the feature (relative contribution of the corresponding feature to the model).

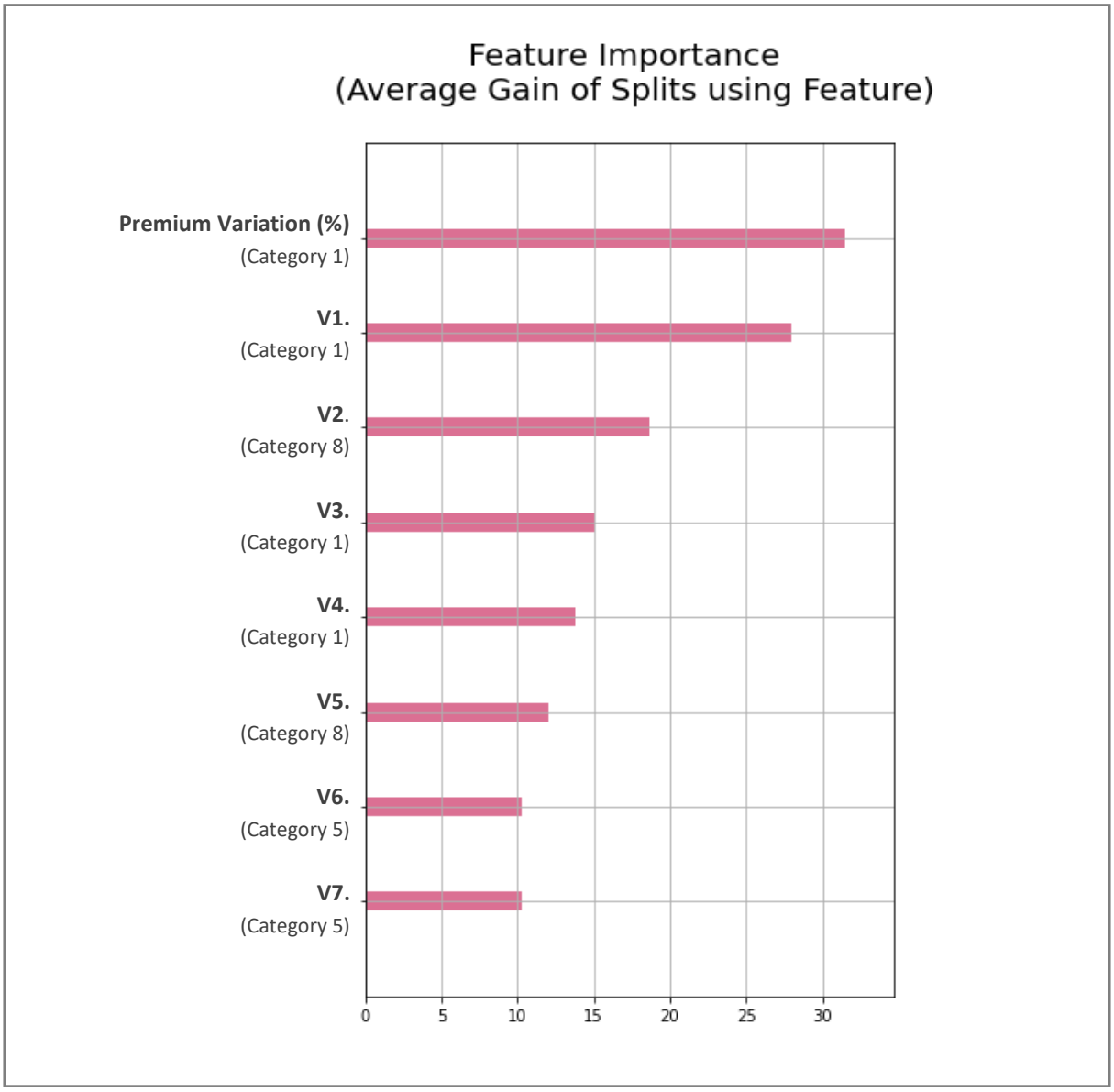


Figure 9.2 Final (Monotonic) Extreme Gradient Boosting model's feature list and respective categories with each feature's importance in final results