NOVA IMS
Information Management School

MEGI

Mestrado em Estatística e Gestão da Informação
Master Program in Statistics and Information Management

IBNR TECHNIQUES IN HEALTH INSURANCE: A MACHINE LEARNING APPROACH

Catarina Ferreira de Jesus de Sousa

Project work presented as partial requirement for obtaining the Master's degree in Statistics and Information Management

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

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

IBNR TECHNIQUES IN HEALTH INSURANCE: A MACHINE LEARNING APPROACH

by

Catarina Ferreira de Jesus de Sousa

Project work presented as partial requirement for obtaining the Master's degree in Statistics and Information Management

Adviser: Professor Doutor Roberto André Pereira Henriques

Co-adviser: Professora Doutora Maria de Lourdes Belchior Afonso

February, 2023

IBNR techniques in Health Insurance: A Machine Learning Approach

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.9.2) Lourenço, 2021.

Acknowledgements

I was entrusted with the opportunity to be an intern for a year to develop this project. It would not have been possible without the vote of confidence that Multicare gave me in the person of Maria do Carmo Ornelas. I would especially like to thank the SAC team for receiving me so well during this time. I take with me a lot of learning from this multidisciplinary team, focused on applying these recent data science methodologies to the business. To Filipa Marques, who was the heart of this project, thank you for your patience, kindness, and vision throughout this development. I could not fail to thank Mariana Vieira, Pedro Lopes, Miguel Cordeiro, and Pedro Gonçalves for all their support. A warm thank you to this incredible team.

Additionally, I would like to sincerely thank Professors Roberto Henriques and Maria de Lourdes Afonso for all your guidance. The areas of specialization of each one were essential to being able to combine machine learning with actuarial expertise in this project. You were tireless in answering all of my concerns.

Finally, I would like to express my gratitude to my family and friends for their support throughout this journey.

Abstract

Loss reserves are typically one of the largest liabilities on an insurer's balance sheet since they can have a significant impact on profits as well as the insurer's solvency. The *Chain Ladder* model is an outstanding actuarial reserving technique that has been applied over the years to estimate Incurred But Not Reported claims.

This project aims to provide the most accurate estimates possible for the calculation and prediction of reserve claim amounts in the context of corporate health insurance. For this, the *Chain Ladder* approach is compared with machine learning algorithms such as the Support Vector Machine (SVM), the Random Forest (RF), the Extreme Gradient Boosting (XGBoost) and Neural Networks (NN).

Keywords: IBNR, Health Insurance, *Chain Ladder*, Machine Learning, Predicting Claims

Contents

List of Figures

List of Tables

Acronyms

R^2	R-Squared 46, 48–50, 52, 54, 55
ANN	Artificial Neural Network 11
CL	*Chain Ladder* 2, 6, 11, 49–52
GLM	Generalized Linear Model 10, 11
IBNR	Incurred But Not Reported 1–3, 8, 10, 39, 55–57
LASSO	Least Absolute Shrinkage and Selection Operator 10
LDFs	Loss Development Factors 6, 10
MAE	Mean Absolute Error 46, 49, 55
ML	Machine Learning 2, 10, 47, 49, 55, 57
MSE	Mean Squared Error 38, 46, 48
NN	Neural Networks vii, 35, 36, 48, 49, 52, 54, 55
RF	Random Forest vii, 31, 32, 38, 42, 49, 51, 55
RMSE	Root Mean Squared Error 46, 49, 55
SSE	Sum of the Square Errors 28
SVM	Support Vector Machine vii, 34, 43, 48, 49, 52, 54, 55
SVR	Support Vector Regression 34, 35
XGBoost	Extreme Gradient Boosting vii, 11, 32–34, 49, 51, 52, 54–56, 58

Chapter 1

Introduction

In health insurance, the insurer covers a specific risk related to a person's healthcare costs. The insurer undertakes to make the agreed payment in case of a random event provided for in the contract in exchange for a premium paid by the policyholder. These events, known as "claims" in the insurance industry, occur almost every day in the healthcare sector.

However, there are several situations where there is a delay between the actual date of the event and the date it is reported to the insurer and accounted for in the balance sheet. This may happen because claims: may be reported within a certain time lag; the claims' settlement process may take a long time or be reopened; or there may be insufficient claim information (Bornhuetter & Ferguson, 1972).

Estimating reserves is then an essential task for any insurer to get an authentic picture of its liabilities, as they are also a measure of a company's financial solvency. On the one hand, sufficient resources are needed to fulfill the liabilities arising from insurance contracts. On the other hand, excess provisions can affect the insurer's profitability.

Loss reserving is one of the most important topics within actuarial sciences. The total loss reserve can be divided into the reserve for known claims and the Incurred But Not Reported (IBNR) reserve. As Skurnick (1973) explains, "the reserve for known claims represents the amount of paid loss that will be required to settle all reported claims not including payments already made on these claims. The IBNR reserve represents the amount of paid loss that will be required to settle all incurred but not reported claims" (p. 17).

A good reserving method will produce an estimated total loss reserve that is close to the required total loss reserve. There is a growing need to select appropriate reserving methodologies and assumptions that can be as practical as they are accurate and are often applied to imperfect data. Insurers must consider the duration of the insurance contract, the kind of coverage provided, and the likelihood of a claim occurring. Insurers also have to adjust their calculations as circumstances change.

Over the past few years, several approaches have been carried out to calculate IBNR.

With the development of artificial intelligence and Machine Learning (ML) models, it has been possible to improve the predictions of these calculations. In partnership with the Portuguese insurance company "Multicare - Seguros de Saúde, S.A." a work project was developed to assess the current method for calculating IBNR and build a new model using machine learning. This project compares traditional models with ML algorithms to understand how accurate each one is when it comes to predicting claim reserves when considering major databases.

Therefore, firstly, the main objective was to understand and measure the level of accuracy that the current methods for calculating IBNR have and, secondly, to build a model using machine learning that could improve the obtained results. That said, this project will mainly focus on the following research question: at any given time, what is the most accurate way to estimate IBNR claims amount? To achieve this, it is necessary to determine if the machine learning model to be developed will be better than the current *Chain Ladder* (CL) method. Other critical research questions will also be considered: Can it be used in real-world business settings? Is it feasible? What are the advantages/disadvantages of ML methods vs. the statistical CL method in forecasting claim reserves?

This master thesis is organized as follows. Section 2 presents the *Chain Ladder* method to give the reader an actuarial background. A brief overview of the work done in this area can be found in Section 3. Section 4 explains the methodology of this project. In section 5 the results are discussed and finally, in sections 6 and 7 the conclusion and recommendations for future work are presented.

1.1 Significance to the insurance company

This project aims to respond to one of the needs of the Corporate Actuarial Support team. The data and conclusions drawn will support the calculations for corporate businesses contracts - when a company offers its employees' health insurance benefits - and not for contracts at the individual level.

When offering health insurance for corporate businesses there are two crucial moments for premium pricing: pricing, when a company wants to purchase a product for the first time and has no history with the insurance company; and renewal which takes place every year and aims to ensure that the insurance company is not losing profits, that is, that premiums are adjusted to the actual aggregated claims.

Following the values of the insurer in question, of "reinventing the past with the future through constant innovation", several projects are underway to optimize pricing and renewal processes using contemporary data science and machine learning methodologies. This project work is one of them, focusing on improving the automation of the IBNR calculation needed for the renewal process.

Each contract has a one-year duration (an annuity). At the beginning of the tenth month, the subscription team begins preparing for the renewal process. Their work depends on the loss ratio (represented in the equation 1.1) of each business.

$$LossRatio = \frac{ClaimsAmount}{Premiums} \tag{1.1}$$

If at the end of the annuity, the loss ratio of a business is greater than 1, that is, the claims amount is greater than the amount received in premiums, then the premium price of that business in the next annuity will have to be higher.

In order to project the future loss rate, it is necessary to be aware of the claims that occurred in the last 9 months, whether those that have already been reported (and therefore are already known by the insurer), as well as those that are yet to be reported (IBNR component). Only by knowing these two installments can the claims that will occur in the last 3 months be projected.

In addition, the calculation of IBNR claims is required every month for the company's accounting. The model built here, if more efficient, should replace the method currently used in order, once again, to automate the process and provide more trustworthy results.

The more reliable these reserve estimates are, the more opportunities the insurer will have to, on the one hand, offer a better product that stands out in the market, with the possibility of having more profit opportunities and, on the other, be more up to date with the current Directive of the European Union, Solvency II, which reduces the risk of insolvency.

Chapter 2

Actuarial Background

2.1 Terminology

The moment in which a claim occurs, known as the accident period, may not coincide with the moment in which the same claim is reported or settled. The time that elapses between these two moments is called the development period. This information is usually represented in *run-off* triangles, as the one shown in table 2.1, that is, an upper triangle where the lines represent the accident period and the columns the development period. These periods can be years, semesters, months, or other.

Table 2.1: Run-off triangle

Accident period i/ Development period j	**0**	**1**	**2**	**...**	**j**	**...**	**n**
0	$X_{0,0}$	$X_{0,1}$	$X_{0,2}$	**...**	$X_{0,j}$	**...**	
1	$X_{1,0}$	**...**					
⋮	⋮						
i	$X_{i,0}$						
⋮	⋮						
n − 1							
n							

Let observations X_{ij}, with $0 \leq i \leq n$ and $0 \leq j \leq n-i$, be the claims amount paid to the insurer for the development period j corresponding to the accident period i. This information, available in table 2.1, is presented in the form of incremental data. It can also be represented in the form of cumulative data through the sum of incremental values:

$$C_{ij} = \sum_{k=0}^{j} X_{i,k} \tag{2.1}$$

The main objective is to estimate the filling of the lower triangle, that is, to infer on the amounts that will be paid in the future.

2.2 *Chain Ladder* Method

The *Chain Ladder* method is one of the oldest actuarial techniques and the one most applied by insurers in estimating the provision for claims due to its simplicity and easy understanding. It is a deterministic model as the estimates obtained are based only on observed historical data and do not assume any probability distribution. In this way, point estimates are obtained, that is, estimates that do not inform about their variability or about their errors. In order to obtain those, Thomas Mack improved this model as will be seen in the next chapter.

The CL model assumes that future loss development patterns will be in line with historical loss development patterns. It is thus based on a set of ratios that relate the amounts for a given year to the amounts for the following year, known as link ratios.

The following equations can be found in Thomas Mack's articles on *Chain Ladder* (1993).

2.2.1 Methodology

For the calculation of reserves by the *Chain Ladder* method, it is necessary to calculate the so-called Loss Development Factors (LDFs) between successive periods of development:

$$\hat{f}_j = \frac{\sum_{i=0}^{n-j-1} C_{i,j+1}}{\sum_{i=0}^{n-j-1} C_{i,j}}, \qquad with \quad j = 0,...,n-1 \tag{2.2}$$

Through the product of the development factors the projection factors are obtained:

$$\hat{F}_k = \prod_{j=0}^{k} \hat{f}_j, \qquad with \quad k = 0,...,n-1 \tag{2.3}$$

Based on the last known accumulated amounts and estimated development coefficients, the following estimate is available to fill the bottom triangle:

$$\hat{C}_{i,j+1} = \hat{F}_j C_{i,j} \qquad with \quad i+j > n; \quad i = 0,...,n; \quad j = 0,...,n-1 \tag{2.4}$$

In this way, it is possible to calculate reserves per year of origin, $\hat{R}_i$, through the difference between the estimate of the accumulated quantity of the last year of development - known as ultimate claim amount - and the last observed value of it:

$$\hat{R}_i = \hat{C}_{i,n} - C_{i,n-i} \qquad with \quad i = 0,...,n \tag{2.5}$$

The total reserve $\hat{R}$, that is, the total expected value of liabilities for claims still outstanding, is calculated as the sum of reserves per accident year $\hat{R}_i$:

$$\hat{R} = \sum_{i=1}^{n} \hat{R}_i \tag{2.6}$$

2.2.2 Model Assumptions

In order to determine if it is appropriate to apply the model to the given data, the model takes into account some assumptions. If one or more assumptions are rejected, the possibility of not applying the model should be considered. The assumptions are as follows:

2.2.2.1 Proportionality between development years

There are development factors, f_j, $j = 0,...,n-1$, such that

$$\mathbb{E}[C_{i,j+1}|C_{i,0},...,C_{i,j}] = C_{i,j}f_j, \quad i = 0,...,n \tag{2.7}$$

The estimates of these coefficients, as seen earlier, are given by the expression 2.2. This assumption implies that individual developmental factors are not correlated.

2.2.2.2 Independence between accident years

The random variables $C_{i,j}$ from different accident years are independent, that is

$$\{C_{i,0},...,C_{i,\infty}\} \text{ and } \{C_{j,0},...,C_{j,\infty}\}, i \neq j, \text{ are independent} \tag{2.8}$$

which leads to the estimators of the development factors, $\hat{f}_j$, $j = 0,...,n-1$, being centered, that is, unbiased, a property that an estimator should always have.

2.2.2.3 Development factor estimators correspond to minimum variance estimators

There is a constant of proportionality, $\sigma_j^2 \geq 0$, such that

$$Var(C_{i,j+1}|C_{i,0},...,C_{i,j}) = C_{i,j}\sigma_j^2, \quad i = 0,...,n; \quad j = 0,...,n-1 \tag{2.9}$$

For each development year j, there is a unique proportionality constant σ_j^2. Thus, it is guaranteed that the estimators of the development factors are the ones with the lowest variance.

Once again, the proof of the results of the previous assumptions can be found in Thomas Mack's articles (1993).

2.2.3 Limitations

In general, this methodology presents eligible results when there is a relatively stable pattern of loss development and a relatively large number of reported claims. It is appropriate for insurers in a relatively stable environment where there are no major organizational changes for the insurer and when there are no major external environmental changes.

However, it is not very robust as it only takes into account arithmetic averages over rows and columns, which makes it highly sensitive to small changes in the data: a small fluctuation in observed data can cause large fluctuations in the estimates. Nevertheless, a singular event, such as an unusually large claim, shouldn't affect an insurer's estimate of IBNR, underlining the need to apply more powerful methods.

It does not include any risk theory or any calendar year effects. It only gives estimates dependent on the upper triangle.

Chapter 3

Literature Review

3.1 Traditional Methods

Within the traditional methods there are other alternatives such as Bornhuetter-Ferguson method, introduced by Bornhuetter and Ferguson (1972). Rather than being based on past experience like the *Chain Ladder* method, it relies on the insurer's exposure to loss. It is most useful when there is a low frequency of claims but with high severity and smooths the variance when there are random fluctuations or major claims at early maturities.

In the late 1980s stochastic models were introduced and since then they have been the subject of several studies. Thomas Mack (1993) developed a stochastic model that produces the same estimation of provisions as the *Chain Ladder* method with the advantage of being able to deduce estimates from the mean squared error, which is associated with the variability of the results. For point estimates, confidence intervals are thus obtained (T. C. Mack, 1999) assuming that the data does not follow a specific probability distribution (distribution-free). Understanding the variability of *Chain Ladder* reserve estimates contributes to valuable information and helps the insurer define which strategy to follow. And one of the biggest advantages of stochastic reserve models is the availability of measures of accuracy of reserve estimates. England and Verrall (2002) developed a comparison between a wide range of stochastic reserving models for use in general insurance such as the Generalized Linear Model, the Generalized Additive Model or the Markov Chain.

All case studies presented in the literature have a common point: it is not possible to guarantee a predominant model, i.e., a model that can be used in any situation. Any insurer must explore existing methods and try to find those that are best suited. Despite their popularity, these traditional methods rely only on the historical value of claims and do not consider other important variables that are available to the insurer and, on the other side, they are estimated by elementary arithmetic procedures, such as averaging over rows and/or columns.

3.2 Machine Learning Methods

The development of areas related to artificial intelligence and machine learning could give a contemporary response to this issue since they provide a sophisticated and efficient tool for understanding and modeling the characteristics of those mentioned objections. Over the years, several studies have been carried out on the application of ML in forecasting loss reserves. Although the IBNR does not correspond to the totality of the loss reserving problem, in the literature is more common to find models adapted to the loss reserves.

There are studies that instead of building complex models, seek to apply ML in improving the traditional methods. John (2018), for example, proposed a claim segmentation with the K-means algorithm before the application of the *Chain Ladder*. As this traditional algorithm only works with aggregate claims, actuaries may have some difficulties in explaining the behavior of a certain estimate. By segmenting aggregate data appropriately into homogeneous segments of data, the interpretation of the results becomes more accessible. On the other hand, forecasts are expected to improve as each claims segment will be able to adopt LDFs more suited to each behavior. However, this methodology did not guarantee the improvement of the results in all cases.

Nevertheless, the main advantage of highly developed computing power in this context is the ability to develop models at an individual level rather than handling aggregated values. This also allows to incorporate other individual features about the insured persons. Wang, We and Qiu (2021) used the data from a health insurance company to explore the effect of adding individual information to loss reserving problems. They compared individual models with and without individual features such as age, gender, policy type, and geometric region, concluding that adding these variables can contribute to more accurate projections of a portfolio's outstanding liabilities. In addition, they drew other interesting conclusions that would not exist without the information about insured persons. For example, older ages tend to increase claims, and claims rates are higher in general health insurance than in critical illness insurance, while in the latter the delays are greater. Although the data to be studied in this project may not follow the same behavior, it is possible to understand the impact of adding information with this level of granularity, thus being an approach to consider when building the database.

Several machine learning techniques have been addressed in recent years, from the simplest to the most sophisticated. McGuire, Taylor and Miller (2018) applied a Least Absolute Shrinkage and Selection Operator (LASSO) regression, which produces a similar Generalized Linear Model (GLM) but at much less time and cost. They used a database of motor injuries with 139.000 claims and variables such as injury severity score, legal representation, operational time and the proportion of claims per quarter. The LASSO regression was tested with reasonable success, managing to model features that are awkward for traditional approaches. However, it tends to perform worse when

there are a large number of predictor variables, and furthermore, it can be difficult for an unsupervised model to recognize some unusual type changes.

Kotsalo (2021) has investigated if machine learning methods can provide better estimations of loss reserves compared to the *Chain Ladder* method. The problem was divided into two: one to predict the claims amount and another to predict the development month. A Ridge regression was applied to predict the claims amount through features such as gender, the policy period, the accident year, the delay in the reporting and settlement days, and the corresponding payment, finding that the CL performs better. However, it is necessary to understand that the dataset used had only 8.000 claims with values below €200, which makes it easier for a simple statistical method to predict. The *Chain Ladder* does not predict the development month question, which might be a piece of valuable information for the insurance company. To predict the month where the claims will be paid Kotsalo used a Logistic regression and Random Forests, where it was concluded that the Logistic regression might be accurate enough to base future decisions. In addition to all this, the two algorithms were not combined which is not very useful in practical terms.

Duval and Pigeon (2019), on the other hand, chose to compare GLMs and gradient boosting models, using a database with more than 67.000 claims considering the variables mentioned above and also the number of health service providers. They concluded that using a model-based only on GLMs could be unstable for loss reserving. But an approach where a gradient-boosting model is applied represents an interesting approach for an insurance company. A gradient boosted decision-tree model requires little data preprocessing and is known for strong performance for prediction on structured data. Particularly, the XGBoost algorithm has a relatively short calculation time. However, the gradient boosting models presented in this paper only allow to compute a prediction for the total paid amount of each claim and not for the payment date - the development month.

Neural Networks can also be a path to follow. Mulquiney (2006) modeled the expected size of finalised claims at each future finalisation date, where it was concluded that an Artificial Neural Network (ANN) model, although more difficult to interpret, resulted in better predictive accuracy compared to a GLM model. This paper represented ANNs through a dataset from a motor bodily injury portfolio with approximately 60.000 claims.

It is now necessary to choose from the literature the best method to answer the question under study. The aim is to find a model that, in addition to being the most suitable possible, is also simple to be implemented.

Chapter 4

Methodology

4.1 Introduction

One of the main objectives of this project is to build a machine learning model to optimise the calculation of claim reserves. However, it is also necessary to explain the assumptions and calculations currently made by the insurer.

Therefore, in this chapter, the methodology used for the traditional approach and for the machine learning approach will be presented. It is important to note that the database to be used in the machine learning models will always have to be in line with the data used in the current model.

Any machine learning strategy should start by following a data mining process. Currently, there is no standard framework to guide the development of data mining projects (Wirth & Hipp, 2000). CRISP-DM (CRoss Industry Standard Process for Data Mining) is a process that provides a framework for carrying out data mining projects that is independent of both the industrial sector and the technology used (Wirth & Hipp, 2000), consisting of 6 phases (Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment) (Chapman et al., 2000).

The methodology chapter will follow some of the phases of the CRISP-DM process represented in figure 4.1. The remaining phases are presented in the remaining chapters. It is essential that this chapter provides the necessary tools to make a comparison between models easily understandable.

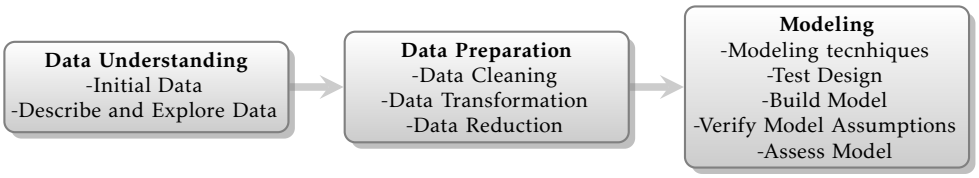

Figure 4.1: Methodology structure

4.2 Data Understanding

The construction of the database was done using the SAS Enterprise Guide software (version 7.1). The ultimate goal was to create a database that included the same inputs as the *Chain Ladder* method, such as claim amounts, accident and accounting dates, but also information at the insured person level. Therefore, the database will need to be organized by company. Given the large amount of data and the computing power available, in order to make this project viable in real time, only a sample of 18 companies was considered. Furthermore, the time span was also reduced. Claims that occurred between 2015 and 2019 were selected - this means that when talking about accounting dates, the year 2020 and the first six months of 2021 were also considered. This selection of data enables a more comprehensive study of several models without the factor of computational power being detrimental.

4.2.1 Collect and Describe Initial Data

To answer the purpose of this project, it was necessary to collect data from different sources. The first one had information about claims, containing 8 variables and 8.586.627 rows. Each observation in this dataset corresponded to a claim for a particular insured person. That person's ID, the claim ID, the date it occurred, the date it was reported, and other information about the claim were therefore available. The variables of the claims dataset are represented in table 4.1.

Table 4.1: Variables Claims dataset

Variable	**Description**
SINISTRO	Claim ID;
CLIENTE	Insured person ID;
NEGOCIO	Corporate business ID;
COBERTURA	Insurance coverage;
DT_EFEITO_SINISTRO	Accident date (format YYYYMM);
DT_CONTABILIZACAO	Accounting date (format YYYYMM);
IDENTIFICA_SIN	Identification of the type of claim;
VALOR_PAGO	Total claims amount (in euros) that the insurer had to pay.

The second one had information about insured persons. One row represents information about an insured person at a particular company in a particular annuity. Therefore, if the same person is insured for, for example, three annuities, it will appear three times in this database. That person's ID and other information about the person is available in about 4.229.046 observations. The 10 variables of the insured persons dataset are represented in table 4.2.

Table 4.2: Variables Insured Persons dataset

Variable	Description
CLIENTE	Insured person ID;
NEGOCIO	Corporate business ID;
COBERTURA	Insurance coverage;
LIMITE_DESPESAS	Available plafond that the insured person has in the respective coverage;
PERC_COMP	Percentage of co-payment that the insurer has to pay if the claim is in reimbursement (for the respective coverage);
ANUIDADE	Annuity start date (format YYYYMM);
PARENTESCO	Degree of kniship;
IDADE	Age;
SEXO	Gender;
DISTRITO	District where the insured person lives.

4.2.2 Data Exploration

In this task, the data is examined more closely to get to know the data beyond its meaning, detect signs of data quality problems, and establish the data preparation steps. For this, simple data manipulation and basic statistical techniques are used. For each variable, the ranges of values and their distribution are analyzed. Some graphs are also presented in the data visualization section as it is essential to understand how the data behaves and connects with each other.

4.2.2.1 Quantitative Data

From the Claims dataset, only the VALOR_PAGO variable is considered as quantitative data. This will be the target variable, since the objective is to predict the amount of money that the insurer will have to pay in the future. As can be seen from the table 4.3, it can assume any value and is therefore a continuous variable. Its distribution is difficult to visualize since the range of values it can assume is very large and most of these values are concentrated close to 0. About 95% of the more than 8 million observations are between 0 and 100 euros. Figure 4.2 depicts the distribution of these 95% of observations. More than 2.7 million are in the first range between 0 and 5 euros. The ranges between 5 and 30 euros are also quite representative.

Table 4.3: Quantitative data description

Variable	Mean	Std Dev	NAs	Median	Min	Max
VALOR_PAGO	54,9	439,6	-	14,0	0	118.645,7
LIMITE_DESPESAS	44.630	197.436	6.857	750	0	1.000.000
PERC_COMP	56,0	33,5	323.265	65	0	100
IDADE	32,7	15,9	-	35	0	111

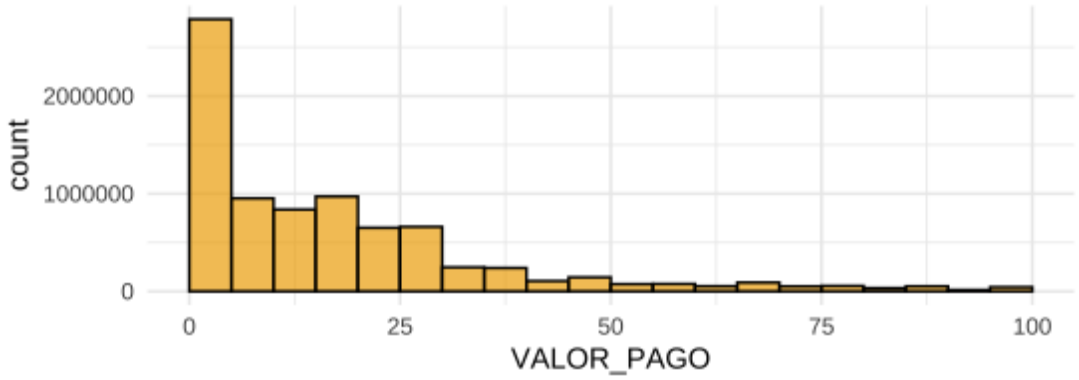

Figure 4.2: Distribution of amount paid (between 0 and 100 euros)

Although age (the IDADE variable) is technically continuous, it is regarded as a discrete variable because the values are always presented as integers. The same happens with LIMITE_DESPESAS and PERC_COMP. The available plafond is always shown in multiples of 10. The percentage of co-payment is an integer value between 0 and 100%. The visualization and interpretation of these variables will be done later in conjunction with other qualitative variables.

4.2.2.2 Qualitative Data

Table 4.4 shows the levels of each qualitative variable. The first, COBERTURA, corresponds to the seven segments that the health insurer covers. It is common to both the Insured Persons and Claims datasets, given that a person has health insurance for certain coverages and the claims that reach the insurer are from those coverages.

Table 4.4: Qualitative data description

Variable	Type	Levels
COBERTURA	Nominal	AMBULATÓRIO; ESTOMATOLOGIA; INTERNAMENTO; MEDICAMENTOS; OUTROS; PARTO; PRÓTESES E ORTÓTESES
DT_EFEITO_SINISTRO	Ordinal	Between 201501 and 201912
DT_CONTABILIZACAO	Ordinal	Between 201501 and 202106
IDENTIFICA_SIN	Nominal	SPE; SPNE; PROV
PARENTESCO	Nominal	TITULAR; CONJUGE; FILHO(A); OUTRO
SEXO	Nominal	F; M
DISTRITO	Nominal	AVEIRO; BEJA; BRAGA; BRAGANÇA; CASTELO BRANCO; COIMBRA; FARO; GUARDA; LEIRIA; LISBOA; PORTALEGRE; PORTO; SANTARÉM; SETÚBAL; VIANA DO CASTELO; VILA REAL; VISEU; ÉVORA; ACORES; MADEIRA; ESTRANGEIRO
ANUIDADE	Ordinal	Between 201501 and 201912

In figure 4.3 it is possible to see how many people are insured for each coverage (in yellow) and how many claims there were for each coverage (in gray). Most people have access to Outpatient (AMBULATÓRIO), Stomatology (ESTOMATOLOGIA), Inpatient (INTERNAMENTO), Prostheses and Orthoses (PRÓTESES E ORTÓTESES) and Other coverages (OUTROS). The number of claims in Outpatient and in Stomatology is much higher. This happens because each person has more than one claim in these coverages. For example, within the Outpatient, a person can go to the emergency room, can go to a doctor's appointment, can do clinical analyses, exams, and many other acts. The number of Inpatients is much lower than the number of people insured.

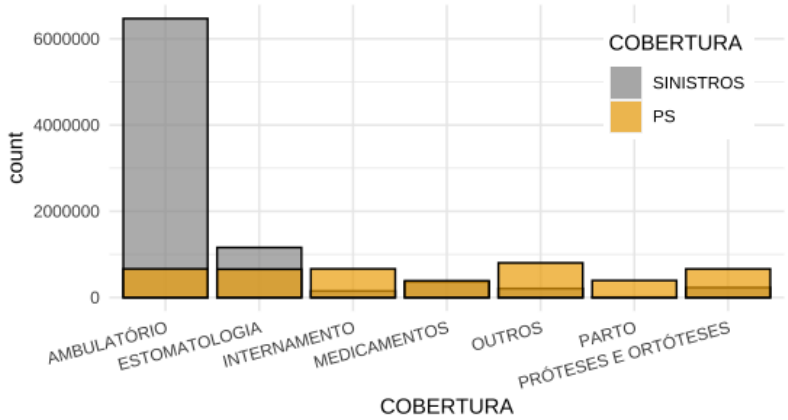

Figure 4.3: Count of coverages by Claims (gray) and Insured Persons (yellow)

A claim can be identified as one of three categories: SPE, when a claim occurs; PROV, when there is some type of authorization; and SPNE, when the claim changes from SPE to PROV, that is, it already has authorization but will still occur. Figure 4.4 shows the distribution of each type of claim in the claims database. The SPEs are the most representative, as they are the most frequent, with more than 60%. Claims in PROV are those that exist in lesser quantity, since the process of accepting a claim is fast. The fact that it takes place after authorization may take longer, and therefore, there is greater representation in SPNE.

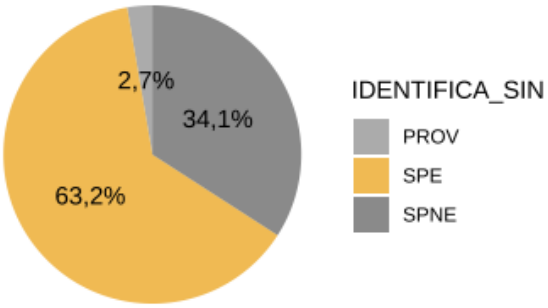

Figure 4.4: Distribution of types of claims in the Claims database

Regarding gender, figure 4.5 shows that there are more women than men in the database of Insured Persons. Most of them are holders (TITULAR), that is, they are employees who work in the companies under study. The rest belong to their household, with children (FILHO(A)) having more representation than the consorts (CONJUGE).

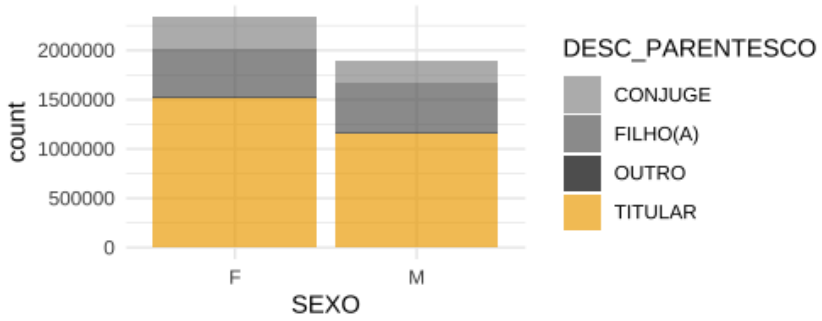

Figure 4.5: Count of kinship degrees (DESC_EPARENTESCO) by gender (SEXO)

Figure 4.6 represents the distribution of people across the districts of Portugal, the islands and abroad (DISTRITO). As the population patterns observed in the country, about 43% of the population under study lives in Lisboa. The districts of Porto and Setúbal also have some representation, with 23% and 11% of the sample. This is due to the fact that most companies are located in the metropolitan areas of Portugal. Other districts that are close to these, such as Santarém, Braga and Aveiro also have some insured persons. The cities that are further inland, such as Portalegre, Guarda and Bragança and the Açores islands, have less population, less employment, and therefore less representation of people who have health insurance. Those who live abroad are few cases where holders live in Portugal.

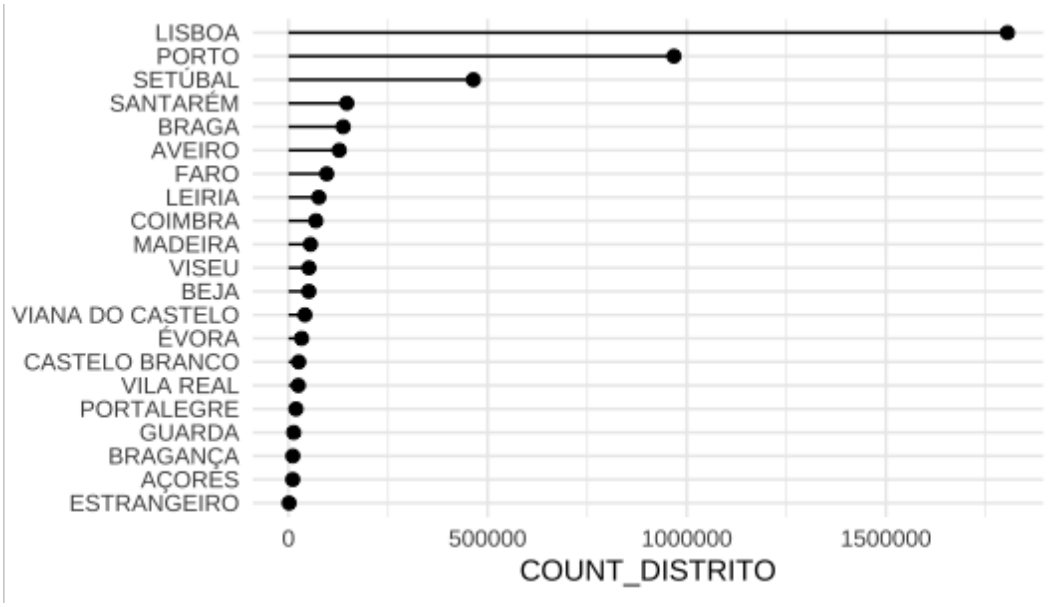

Figure 4.6: Count of Insured Persons by district

4.2.2.3 Data Visualization between different types of variables

Figure 4.7 represents the age distribution (IDADE) by gender (SEXO). Given that the insurance contracts chosen for this project cover not only holders but their household, it is to be expected that children (age 15 and under) will have some representation in this database. It is normal that the number of young people between the ages of 15 and 20 is not so high given that it is an age when many lose their parents' health insurance. On the other hand, there are also not many holders under the age of 20. The remaining distribution follows a normal behavior, with significantly more women than men.

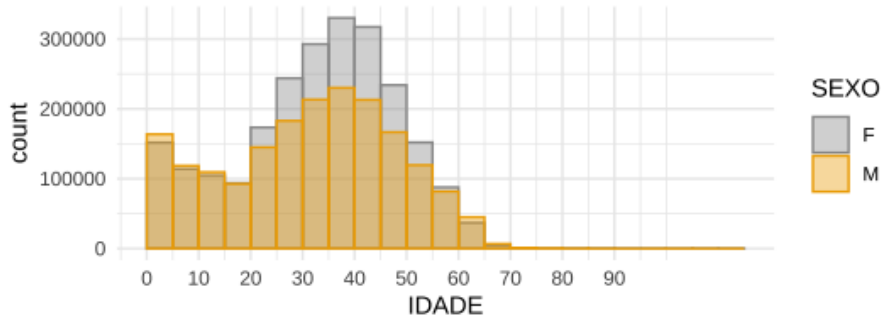

Figure 4.7: Age distribution (IDADE) by gender (SEXO)

Other important variables are the plan conditions to which each person has access. In this case, we have information on the ceiling (LIMITE_DESPESAS) that can be spent on each coverage (figure 4.8) and the percentage of reimbursement (PERC_COMPARTICIPARCAO) that the insurer pays (figure 4.9). The higher the ceiling and the higher the percentage of co-payment, the more expenses will be borne by the insurer.

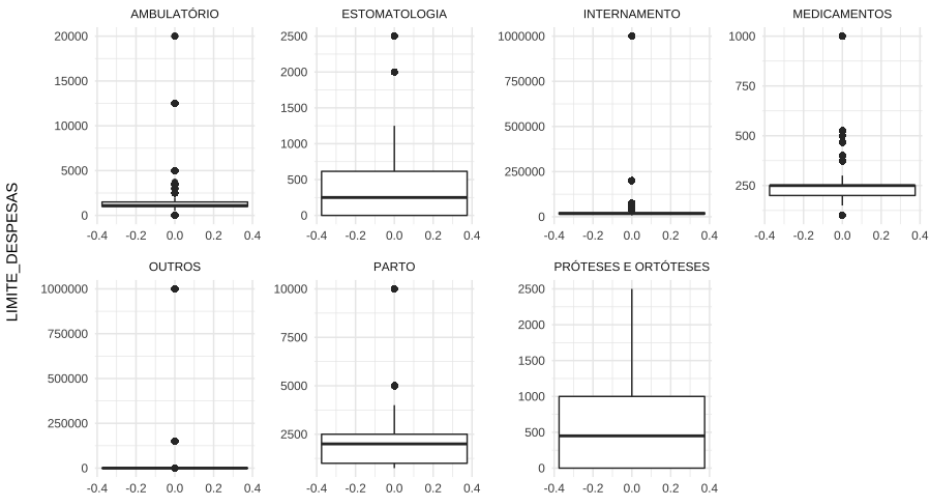

Figure 4.8: Plafond boxplot by insurance coverage

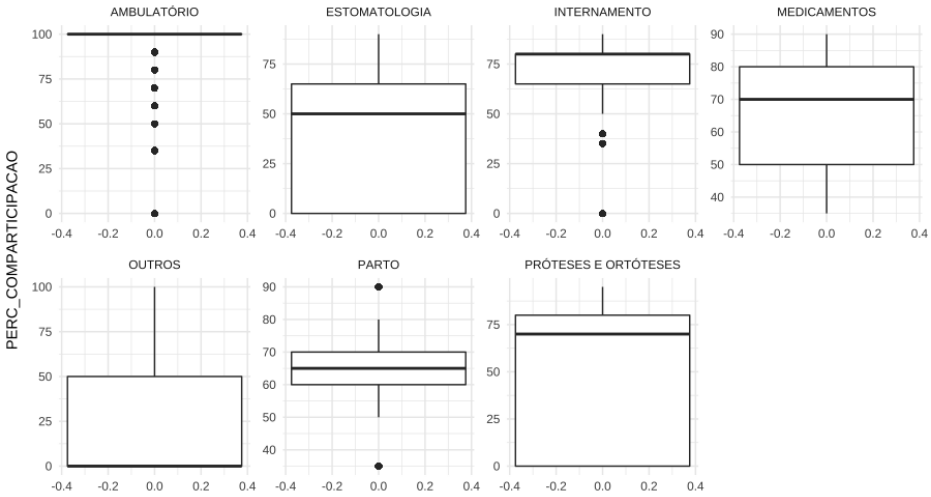

Figure 4.9: Reimbursement percentage boxplot by insurance coverage

For an Outpatient, on average, people have 1.000 euros available and the insurance company usually pays all expenses (which does not happen in any other coverage). Coverages such as Stomatology, Medicine, Childbirth, and Prostheses and Orthoses have a lower available plafond between 250 and 2.000 euros with reimbursement between 50 and 75%. In the case of Inpacient, the available plafond varies a lot depending on the company. However, on average, people have 100.000 euros available and the insurance company pays an average of 80% of the expenses.

To end the exploratory data analysis, figure 4.10 shows the total amount paid (over the 6 years available) that the insurer had to pay per company. As mentioned before, in this database there are 18 businesses that are represented here with the letters of the alphabet. Businesses A, B, and C are the ones with the highest expenses (since they are probably the ones with the most insured people). Inpatient and Outpatient care are the most significant coverage in all cases.

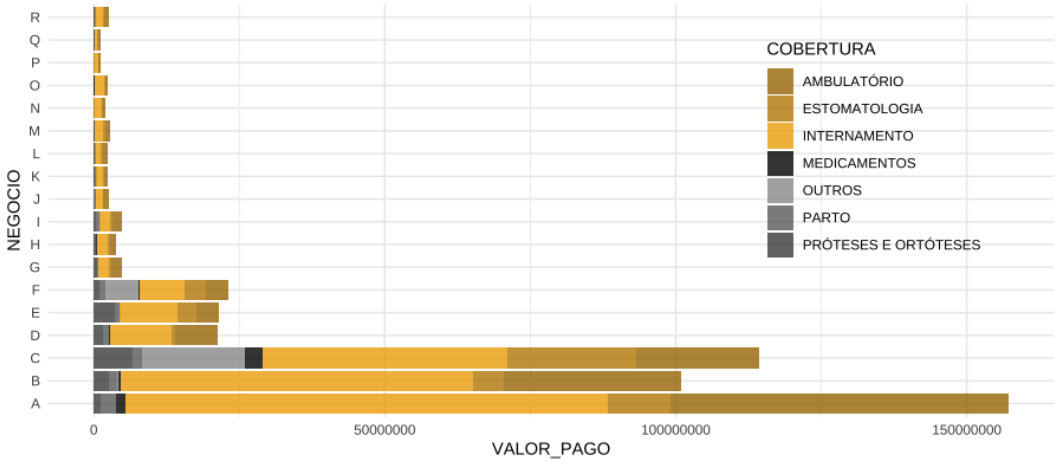

Figure 4.10: Paid amount by insurance coverage (COBERTURA) per business (NEGOCIO)

4.3 Data Preparation

Data preparation is the process of cleaning and transforming raw data. This phase encompasses all operations required to create the final dataset, which will be used to feed the models. For the sake of this project, this chapter will begin by transforming the data so that later on, it's possible to integrate the two initial databases. The cleaning and selection of the most relevant variables will take place only after that. From these last 2 processes, the software used became R Cran, version 4.2.0.

4.3.1 Data Transformation

4.3.1.1 Aggregate data

As explained earlier, the goal is to create a database that includes the same inputs as the *Chain Ladder* method. Therefore, the information at the level of the insured person had to be structured by business, for a certain year and month of accident, for a certain year and month of accounting, and for a certain coverage. As a result, the data was aggregated.

In the case of the Claims dataset, the variables were grouped by: business ID (NEGOCIO); coverage (COBERTURA); accident date (DT_EFEITO_SINISTRO); accounting date (DT_EFEITO_CONTABILIZACAO); and type of claim (IDENTIFICA_SIN). The amount paid is now summarized as the sum of the total amount paid for each of these groups (SUM_of_VALOR_PAGO for now referred to as VALOR_PAGO_TOTAL). Thus, the claim ID (SINISTRO) and the client ID (CLIENTE) dropped from the database, as they correspond to individual variables.

The similar approach was used to the Insured Persons dataset. The data was grouped by: business ID (NEGOCIO); coverage (COBERTURA); and annuity start date (ANUIDADE). The client ID variable also dropped. The rest of them were summed up as follows:

- **LIMITE_DESPESAS**: Average plafond for that business, coverage and annuity;
- **PERC_COMP**: Average percentage of co-payment for that business, coverage and annuity;
- **PARENTESCO**: Total count of people of each kinship, (for that business, coverage and annuity), thus creating 4 new columns - total count of holders, children, consorts and others;
- **IDADE**: Variable used to create 8 new columns. For that business, coverage and annuity - the average age, the total count of people with less than 18 years, the total count of people with more than 65 years, the 10th, 25th, 50th (median), 75th and 90th percentiles of the distribution of each population. The idea behind the percentiles is that the models could better capture the age distribution.
- **SEXO**: Total count of people of each gender (for that business, coverage and annuity), thus creating 2 new columns - total male count and total female count;

- **DISTRITO**: Total count of people living in each district (for that business, coverage and annuity), thus creating 21 new columns - each corresponding to a level of the DISTRITO variable.

4.3.1.2 Construct data - before data merging

ANO_MES While the Insured Persons dataset only has information on the first month of the year the annuity started, the Claims dataset contains both the month and year of accident. There is a need to have 12 rows for each annuity instead of just 1. Therefore, a simple dataset was created with the MES variable, with 12 rows, each filled with the numbers 01 to 12 (representing January to December). This new dataset was merged with the dataset of the insured persons, with a join by no variable - meaning that each row would be multiplied by 12.

The first 4 characters were selected from the ANUIDADE variable, which corresponds to the year in which the annuity began. And they were added to the MES variable to recreate the ANO_MES variable with the condition that the annuity is always respected. For example, an annuity starting in April 2015, will now have an ANO_MES of '201504', '201505', ..., '201512', '201601', '201602', and '201603'.

MES_NEGOCIO The fifth and sixth characters of the ANUIDADE variable were selected, so that there is a variable that indicates the month when the annuity starts (which will be important in the division between train and test). And, instead of 2 characters with numbers, they were replaced by the respective months, ('01' becomes January, '02' becomes February, and so on). ANUIDADE variable drops.

DELAY A key variable is now created: DELAY. It will be one of the links between the two datasets, and one of the most crucial for the models.

It will be built first in the Claims dataset. The delay is the difference, in months, between the accounting date (also known as development date) and the date of occurrence. So, for example, assuming that the DT_EFEITO_SINISTRO is '201501' and the DT_CONTABILIZACAO is '201604', the DELAY will be 15 (months). Thus, the DT_CONTABILIZACAO drops from the database, since all that is needed is the date of occurrence and how many months later it was reported to the insurer. It goes without saying that the delay will be 0 if the claim is submitted in the same month that it occurred. All claims with a delay of more than 12 months were disregarded due to computational constraints and practical considerations. This is due to the fact that the money that must be paid after 1 year is insignificant (less than 1% of the total) and because it will be easier to merge with the other dataset.

The same was done for the dataset of insured persons as it did for the annuity months: a variable named DELAY was introduced, with values ranging from 0 to 12, multiplying each row by 13, making it possible to merge them later.

For the same set of business, coverage, year and month, and type of claim, regardless of the delay, the values of the variables of the insured persons will be the same. The only variable that changes will be the delay, and in the future, the amount paid.

IDENTIFICA_SIN The same thing goes for the type of claims. For each combination of business, coverage, year and month of occurrence and delay, there is a need to have the information about SPE, SPNE or PROV claims. That's why the IDENFICA_SIN variable was added to the Insured Persons dataset, which multiplies each line by 3.

4.3.1.3 Data Merging

The two datasets are then merged into one after having all the essential variables. The join scheme is shown in figure 4.11.

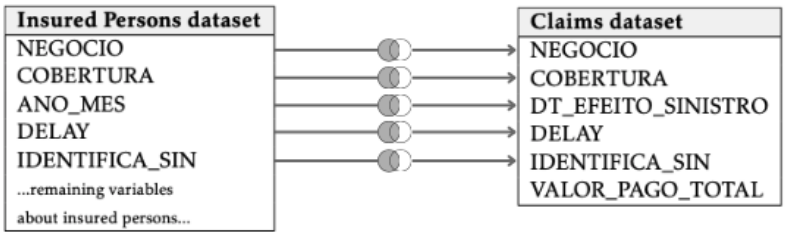

Figure 4.11: Data merging scheme

The reason why there was so much concern over multiplying the Insured Persons dataset also explains why a left join is made in this way, where all of the insured persons' data appears. It's because this project's objective is to determine how much money will need to be paid in the future for claims that have already happened, taking into consideration the characteristics of the insured persons. And for that, just like a time series, it needs all the reference points. There may be combinations of business, coverage, year and month, delay and claim types that will not have a paid amount (appearing as a missing value). But the goal is that even in these cases, the models can predict that the insurer won't have to pay anything.

4.3.1.4 Construct data - after data merging

Separate ANO from MES The year and month are now two different variables.

OUTRO_PARENTESCO This variable becomes the sum of the consorts with the others, given the little significance they have. So the CONJUGE variable drops.

TEMPO_EM_RISCO Sum of the total number of people in that business, for that coverage, year and month of accident. (It's just the sum of columns M and F).

VALOR_PAGO_TOTAL When using the *Chain Ladder* method, the insurer does not take into account the type of claim. This is because the two components SPNE and PROV are cumulative over time. Assuming that a person requests an authorization of 500 euros for a claim in one month and that authorization is accepted. Even if it only refers to one authorisation, if the person does not use it for three months, the 500 euros will appear three times in the database.

Therefore, the variable VALOR_PAGO_TOTAL will, from now on, be the sum of the SPE with the variation of the SPNE between accounting months and the variation of the SPNE between accounting months, as is currently done for the *Chain Ladder* method. So, IDENTIFICA_SIN variable drops from the database.

DELAY0 The claim amount (in euros) that the insurer already paid for that business, coverage, year, and month, when the accident month is the same as the reporting month.

This variable is simply the value of the VALOR_PAGO_TOTAL variable at delay 0. It is important to know the amount of claims that were reported in the same month they occurred, as this could affect the future, that is, the same accident month, with a delay greater than 1.

VP_ACUMULADO The accumulated claim amount (in euros) that the insurer already paid for that business, coverage, year and month.

For example, for business A, for Outpatient coverage, the VP_ACUMULADO in January 2015 delay 4, (that is, the accumulated amount that has already arrived until May 2015), is the sum of the VALOR_PAGO_TOTAL variable in January 2015, for delays 0, 1, 2, and 3. Once again, knowing what has arrived at a given time can be crucial to predicting the future.

RANDOM Random number generated through a normal distribution with mean 0 and standard deviation 1.

When selecting features, these can be ranked in decreasing order of importance to the output. Only the most relevant features, the top ones of the list, should be selected. The question is how to establish the cutoff point between them, i.e., what features should be chosen and what should be eliminated. If a random variable that isn't at all correlated with the output is more significant than other variables in the dataset, it means that these variables will have no value for the models.

4.3.2 Data Cleaning

4.3.2.1 Missing values treatment

As mentioned before, some values of the amount paid variable were assigned with 'NA' as a result of the left join of the insured persons with the claims datasets. This

means that for that business, coverage, year, month, and delay, there were no claims. Therefore, in cases where VALOR_PAGO_TOTAL is missing, it will be replaced by 0.

The PERC_COMP variable follows the same line of thought. If it is missing, the co-payment percentage is 0 because the insurance company does not contribute anything to reimbursement.

The same could be done for the LIMITE_DESPESAS variable: if it was missing, then it would be 0. However, if the plafond that a particular person has to spend on a particular coverage is zero, it means that person does not have access to that coverage. So, when the plafond is missing, this row is eliminated from the database. Most of the rows that were eliminated were from the OTHERS coverage, which does not have a great impact on the variable to be studied. It was also guaranteed that for these deleted lines, there were no values different from 0 in the VALOR_PAGO_TOTAL variable.

At this point, there are no more missing values in the database.

4.3.2.2 Outliers treatment for the target variable

A special consideration is given for the outliers treatment of the target variable, the VALOR_PAGO_TOTAL. It was chosen to handle these since it was not desired that extremely rare events have an impact on the models. Undoubtedly, the goal is to predict any value that might be received by the insurer, but there are rare observations that completely miss business and coverage behaviour. This phenomenon can be seen in the boxplots in figure 4.12.

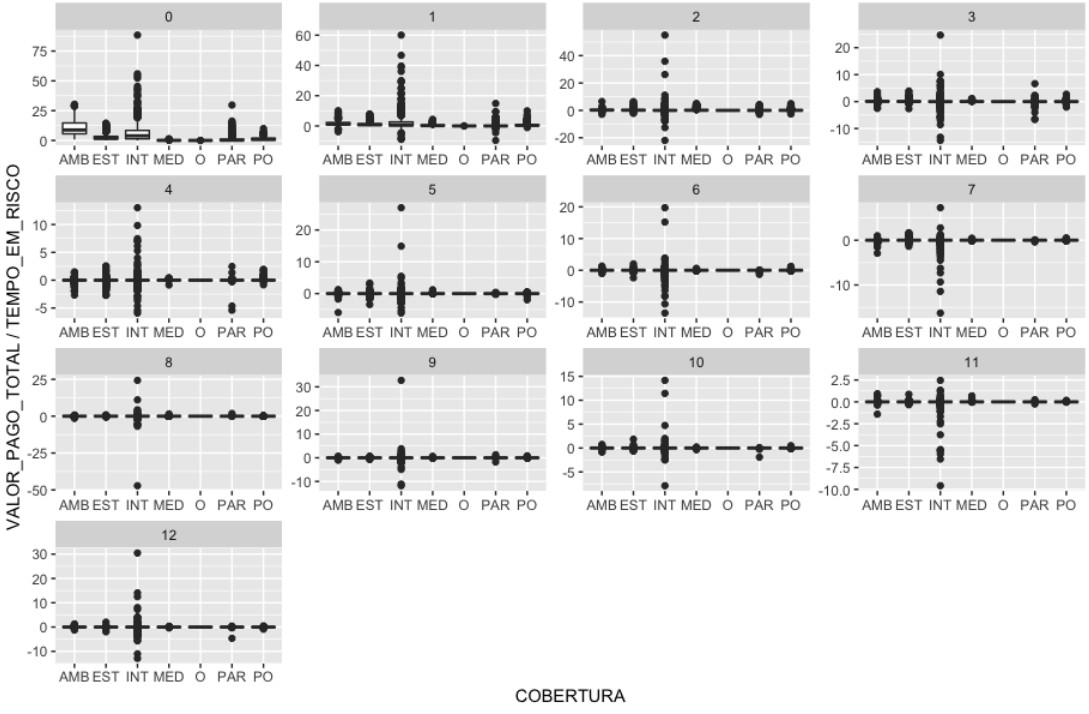

Figure 4.12: Boxplot of relative paid amount for coverage and delay

For this analysis it did not make sense to directly compare the values of the total amount paid given that a company with more employees would incur higher costs. A company with fewer employees, on the other hand, will present lower costs, leading to the misalignment of the outliers. Therefore, the VALOR_PAGO_TOTAL was made a relative measure, as it was divided by the TEMPO_EM_RISCO (represented on the y axis). In this way, the amount paid per insured person is analyzed.

On the other hand, since, for instance, a Stomatology claim that arrives 10 months late cannot be compared to an Inpatient claim that arrives 1 month late, the boxplots are separated by delay (from 0 to 12) and by the 7 available coverages (represented on the x axis).

In order to treat only extreme events, for this project, the 1% and 99% percentiles of each boxplot were considered outliers, which correspond to about 15 to 20 observations per boxplot. These rows would never be deleted from the database as, once again, it would make no sense to lose knowledge from a period of time. Instead, it was decided to replace these observations.

The outliers imputation was done as follows:

1. For each outlier, the business ID to which it belongs, the coverage and the delay were identified. For instance, in figure 4.12 it is clear to see that there is an outlier of value 60 for delay 1, Inpatient coverage. This outlier belongs to business K.

2. Then all the values that have the same characteristics as that outlier were selected. All the values that this business, coverage and delay have in common and that are not outliers. Specifically, values with the same characteristics that happened throughout different year-month combinations. Taking into account the previous example, all values of business K , of Inpatient with delay 1, that are not outliers, would be selected.

3. If the outlier belongs to the 1st percentile, then that value is replaced by the minimum of the values selected in the previous point. If the outlier belongs to the 99th percentile, then it will be replaced by the maximum. The goal is to substitute the outlier with the closest number that is not an outlier but still has the same features. Thus, high values will remain high and low values will remain low, but not with such extreme events. Considering the example, the outlier which actually corresponded to 14.835 euros, was replaced by 2.649 euros, which was the highest value of business K, for Inpatient and delay 1.

4.3.2.3 Coherence checking of the target variable

Figure 4.13 shows the distribution of the total paid amount by delay and insurance coverage. As would be expected, most claims are reported in the month in which they occur, i.e., they have a delay equal to 0. The longer the delay, the lower the number of claims reported, as most are reported up to 3 months later. As is also to be expected,

the largest payments go to Outpatient and Inpatient care. What would no longer be expected is that there are some observations with a negative value paid. This happens because, as mentioned earlier, the VALOR_PAGO_TOTAL variable has become the sum of the SPE with the variation of the SPNE between accounting months and the variation of the SPNE between accounting months. The variation of SPNE and PROV can be negative and make the VALOR_PAGO_TOTAL variable also negative.

In this case, it then makes sense to have negative amounts paid. However, having values with such a huge amplitude and dealing with both positive and negative values is a significant challenge for any model.

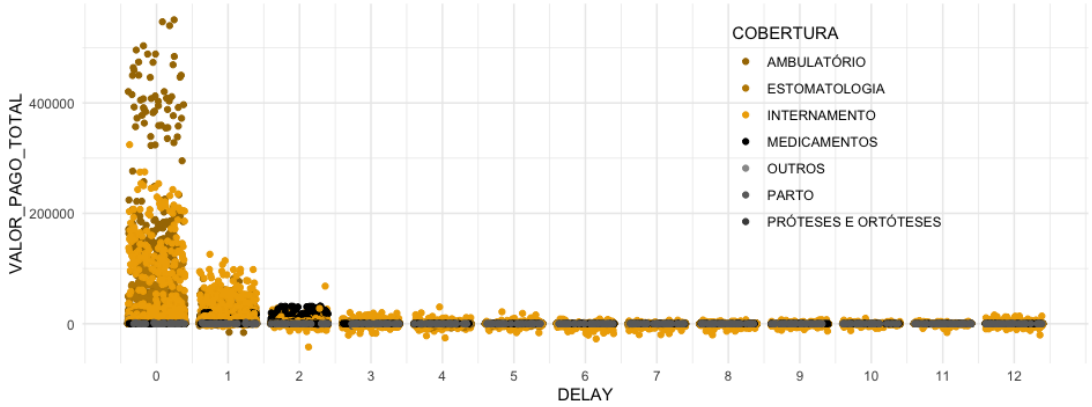

Figure 4.13: Distribution of total paid amount (VALOR_PAGO_TOTAL) by delay and insurance coverage (COBERTURA)

4.3.3 Data Reduction - Clustering District Variables

At this point, the variable 'DISTRITO' has been transformed into columns, each containing the number of people based in each district. Given the size of the dataset and the purpose of the problem, there was no need to have 21 columns indicating how many people there were in each place. Therefore, a clustering algorithm was used to divide the districts into groups with similar characteristics.

The K-Means algorithm was chosen to accomplish this, due to its simplpicity. After defining the k number of centroids, it allocates every observation to the cluster whose centre is closest. The number of centroids was defined by the Elbow method.

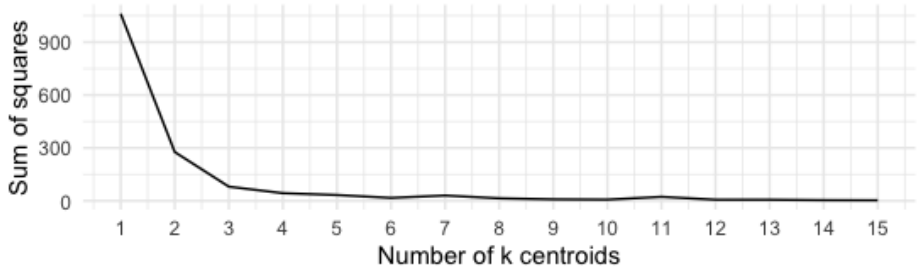

Figure 4.14: Elbow graphic

The Elbow method runs the K-Means algorithm for each k, in a predefined range, and calculates the Sum of the Square Errors (SSE) of each one of them. The goal is to choose a small value of k that still has a low SSE, i.e., a low distance to the centre.

In order for the districts to be comparable, 2 pieces of information were extracted about each one: the average paid amount and the average delay. The Elbow Graphic 4.14 indicates that 3 is the ideal number of centroids. Then, figure 4.15 shows how K-Means divides the districts into the 3 clusters.

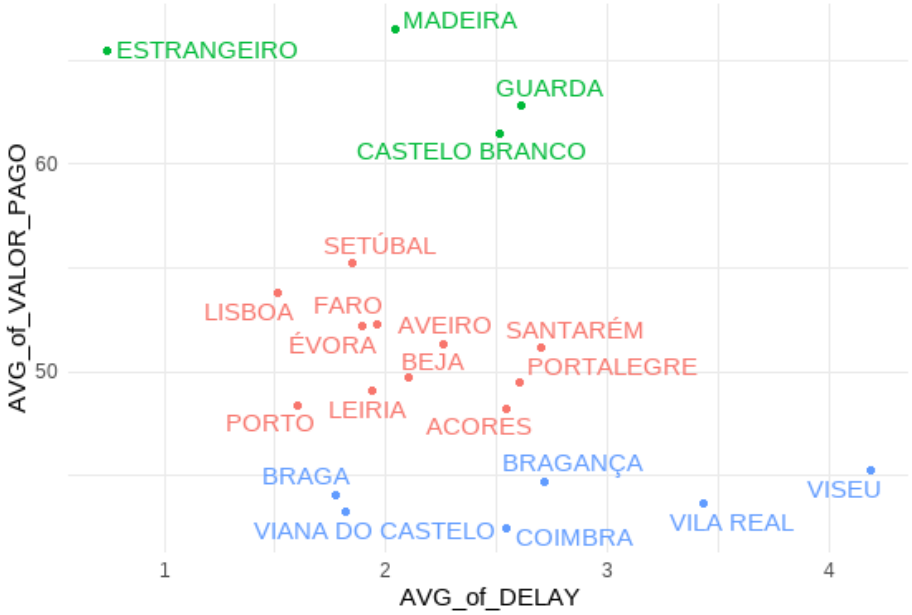

Figure 4.15: Clustering Districts

The clusters are essentially divided by the average paid amount: those that pay the most (green), medium (red) and least (blue). The red cluster is the one that contains the largest number of people, as it includes the metropolitan areas of Portugal. The green cluster has, on average, higher paid amounts because it is likely that in those districts of the country the supply of medical services is lower. Finally, the blue cluster is the one that has the lowest average paid amounts, however, it is also where the average delay to report claims is greater.

4.3.4 Current Dataset

As an overview of everything accomplished in the methodology chapter until this point, the dataset is presented in this section. The current dataset has 71.042 observations. Table 4.5 includes the 30 variables and a brief description of each one. It is divided into sections: the first corresponds to the identification variables; the second corresponds to the variables about the people insured for that business, coverage, year and month (e.g., how many people had business A, for January 2015); the third corresponds to variables with amounts in euros; and the last to a random variable.

Table 4.5: Variables currently available on the dataset

Variable	Description
NEGOCIO	Corporate business ID;
COBERTURA	Insurance coverage;
ANO	Accident year;
MES	Acciedent month;
DELAY	Delay (in months) between the accident month and the development month;
LIMITE_DESPESAS	Average plafond;
PERC_COMP	Average co-payment percentage for the insurer if the claim is in reimbursement;
MES_NEGOCIO	Month that the annuity for that business started;
TITULAR	Total holders count;
FILHO.A	Total children count;
OUTRO_PARENTESCO	Total count of other kniship;
TEMPO_EM_RISCO	Total count of insured persons;
M	Total male count;
F	Total female count;
IDADE_MEDIA	Average age;
IDADE.18	Total count of people under the age of 18;
IDADE.65	Total count of people above the age of 65;
IDADE_PCTL10	10th percentile of the age distribution;
IDADE_PCTL25	25th percentile of the age distribution,
IDADE_PCTL50	Median of the age distribution,
IDADE_PCTL75	75th percentile of the age distribution,
IDADE_PCTL90	90th percentile of the age distribution,
CLUSTER_DISTRITO1	Total count of people living in Açores, Aveiro, Beja, Évora, Faro, Lisboa, Leiria, Portalegre, Porto, Santarém and Setúbal;
CLUSTER_DISTRITO2	Total count of people living in Braga, Bragança, Coimbra, Viana do Castelo, Vila Real e Viseu;
CLUSTER_DISTRITO3	Total count of people living in Castelo Branco, Guarda, Madeira and abroad;
VALOR_PAGO_TOTAL	Total claims amount (in euros) that the insurer had to pay in that year, month and delay;
DELAY0	Total claims amount (in euros) that the insurer had to pay in that year and month (at delay 0);
VP_ACUMULADO	Accumulated claims amount (in euros) that the insurer had to pay in that year and month until that DELAY;
RANDOM	Random numbers from a normal distribution.

4.3.5 Correlation Analysis

In order to discover multicollinearity issues and identify features that may have greater predictive power in a model, it is crucial to check for correlation.

The first step was to calculate the correlation between variables regardless of their type, i.e., whether they are continuous or categorical. To do this, all non-numeric variables were one-hot encoded using the *model.matrix()* function from the *stats* package (version 4.2.0). As no categorical variable showed highly correlated levels, and to make the data easier to visualize, from this point on, only the continuous variables were worked on. Their correlation can be seen through figure 4.16. The Spearman method was used in the calculation because it was not possible to guarantee a linear relationship between the variables.

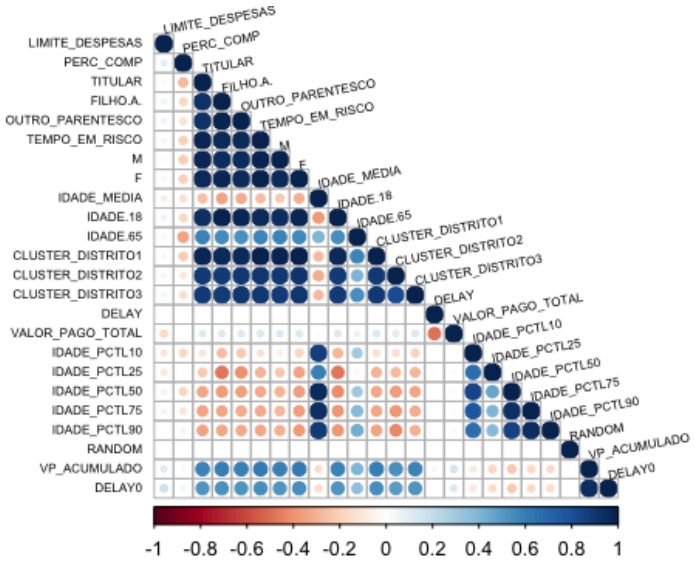

Figure 4.16: Correlation plot between quantitative variables

As would be expected, 'TEMPO_EM_RISCO' variable has a high level of correlations, as it represents the total number of insured persons and:

- 'TITULAR' + 'FILHO.A.' + 'OUTRO_PARENTESCO' = 'TEMPO_EM_RISCO'
- 'M' + 'F' = 'TEMPO_EM_RISCO'
- 'CLUSTER_DISTRITO1' + 'CLUSTER_DISTRITO2' + 'CLUSTER_DISTRITO3' = 'TEMPO_EM_RISCO'

The age variables also present high correlations among themselves. As for the target variable 'VALOR_PAGO_TOTAL', it only highlights the relationship with the 'DELAY', which will certainly be a feature present in the models.

As regression models are not applied in this project, these variables will not be previously removed. Machine learning models can handle multicollinearity. However, a possibility for improving the models is identified here: not taking into account variables that are highly correlated.

4.4 Modeling

In this phase, the chosen techniques are specified. A brief description is given of each model, the assumptions that each requires, and how their parameters could be calibrated to optimal values. The division between training and testing is explained, as is how the most important variables are chosen. In the end, the measures that will enable model comparison and optimum model selection are presented.

4.4.1 Modeling Techniques

It is crucial to identify the kind of problem that has to be solved in order to select the appropriate model. In this project, the objective is to predict the amount that the insurer will have to pay in the future on claims that occurred in the past. It is undoubtedly needed a supervised learning regression algorithm. Supervised learning algorithms try to model relationships and dependencies between the target prediction output ('VALOR_PAGO_TOTAL') and the input features such that it's possible to predict the output values for new data. And, it is a regression algorithm as the objective is to predict continuous outcomes.

4.4.1.1 Random Forest

RF is an ensemble learning method built out of Decision Trees, developed by Leo Breiman (2001). Ensembles use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. In addition to improving accuracy, an ensemble reduces the spread or dispersion of the predictions.

RF algorithm have three main hyperparameters, which need to be set before training. These include node size N, the number of trees K, and the number of features sampled F. The algorithm works as follows:

1. Randomly select K subsets of data from the training set to construct K bootstrap datasets with repeated samples;

2. K trees are built using each bootstrap dataset. Each tree is built until there are fewer or equal to N samples in each node. In each node, F features are randomly selected (F is generally defined as the square root of the total number of features of the original dataset);

3. There are K trained models, and the final result for the regression task is produced by averaging the predictions of the individual trees – a technique known as Bagging (Breiman, 1996);

4. Samples that to not appear are called 'out-of bag' samples and are used to validate the results through cross-validation.

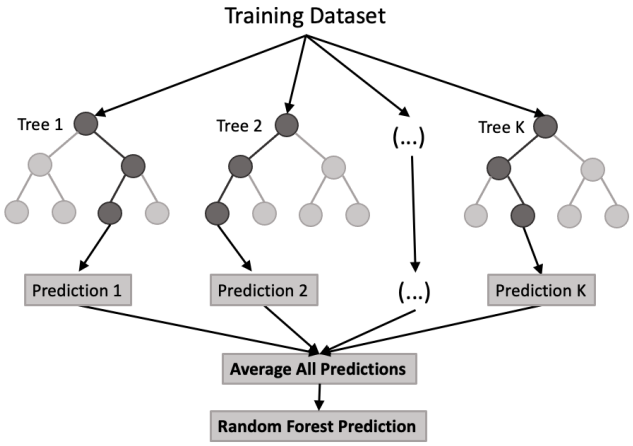

Figure 4.17: Flow chart of a Random Forest Algorithm

Random Forest limits the greatest disadvantage of Decision Trees. Through the Bagging technique, it reduces the risk of overfitting due to subset and feature randomization. Firstly, because it uses a unique subset of the initial data for each model, which helps to make Decision Trees less correlated. On the other hand, it splits each node in every tree using a random set of features, which means that no single tree sees all the data. This helps to focus on the general patterns within the training data, reducing the correlation among decision trees and minimising sensitivity to noise. By averaging uncorrelated trees, the total variance and prediction error are decreased.

RF also provides flexibility. It works well with no hyperparameter tuning, it is rather fast, robust, and can show feature importance. Moreover, it can handle large datasets efficiently and produce good predictions that can be easily understood. All of these possibilities make it a good option when compared to linear models.

This algorithm is not flawless, though. As it often works with larger datasets, it requires more resources to store the data. Besides that, the process might be time-consuming as it needs to compute data for each individual decision tree.

Also, a major disadvantage is that RF is not able to extrapolate based on the data. The predictions it makes are always in the range of the training set. The Random Forest Regressor is unable to discover trends that fall outside of that range. So, if the problem to be solved requires identifying any sort of trend, Random Forest might not be able to formulate it.

4.4.1.2 Extreme Gradient Boosting

While bagging consists of different individual models learning in parallel, boosting involves learning models in a sequence, i.e., each model is created taking into account the previous one. Within the gradient boosted trees algorithms it was developed the XGBoost by Chen and Guestrin (2016). Given that it is an "optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable" as the authors

mentioned, it is currently one of the most popular machine learning techniques. A brief description of its workflow is shown below:

1. The first line of the algorithm initializes to the optimal constant model, which is just a single terminal node tree. The residuals are then computed;

2. According to the additive training strategy of boosting, each tree is constructed based on learning from the residual r of the previous tree. The prediction of the i-th iteration is given by $\hat{y}_i = \hat{y}_{i-1} + F_i(X)$, $i = 1,...,K$. At every iteration, XGBoost optimizes the model and decreases the prediction error;

3. Repeat until the specified number of trees K is reached. The final prediction output $\hat{y}_K$ is generated by the weighted summation of trees as follows:

$$\hat{y}_K = \sum_{i=0}^{K} F_i(X), \qquad F_k \in \mathcal{F}, \tag{4.1}$$

   where $\mathcal{F}$ is the space of functions containing all regression trees;

4. To learn function F of each tree, XGBoost minimizes a regularized (L1 and L2) objective function that combines a convex loss function (based on the difference between the predicted and target outputs) and a regularization term to penalize the model complexity and prevent overfitting.

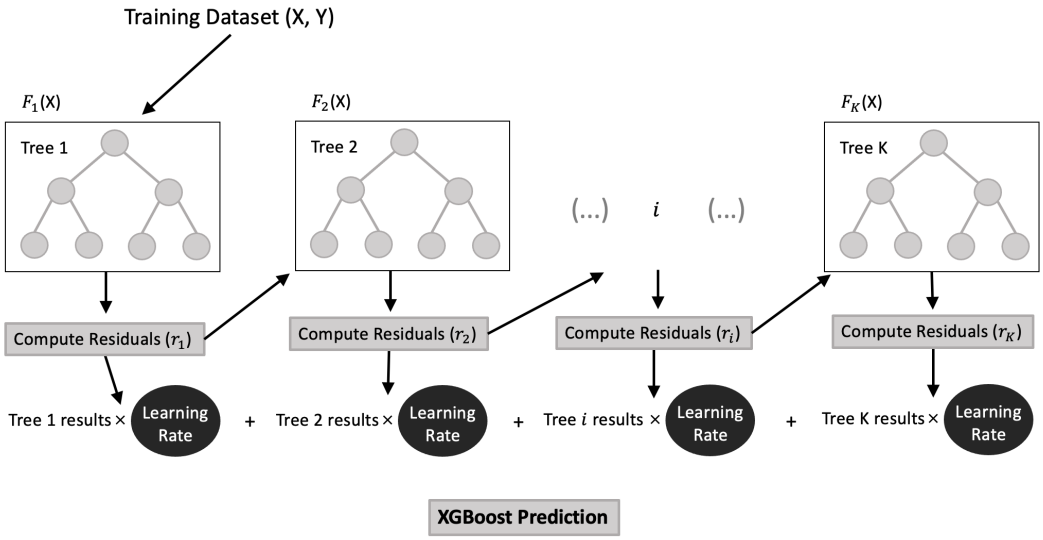

Figure 4.18: Flow chart of an Extreme Gradient Boosting Algorithm

XGBoost is reliant on the performance of a model and computational speed. It provides several benefits, including parallelization, distributed computing, cache optimization, and out-of-core computing. During training, XGBoost uses parallel computation to build trees across all CPU cores. It also distributes computing when it is

training large models using machine clusters. Algorithms and data structures can also benefit from cache optimization to make the best use of the hardware that is available. For larger data sets that won't fit into the conventional memory size, out-of-core computing is used.

Furthermore, it can automatically determine the optimal missing value based on training loss, meaning that it can handle well with missing values. It includes regularization to prevent overfitting; it has in-built cross validation capability; and tree pruning uses a depth-first approach. This greatly enhances XGBoost's computational efficiency and speed compared to competing GBM frameworks or other models.

However, due to its high capacity to identify complex relationships, it is more likely to overfit than bagging techniques do. There are several ways to attenuate this drawback, including specifying some hyperparameters, such as regularization and early stopping. XGBoost might be sensitive to outliers, and it is almost impossible to scale up, because every estimator rests its accuracy on the prior predictions.

4.4.1.3 Support Vector Machine - Regression

SVM is a popular machine learning tool for classification and regression, introduced by Vapnik (1995). The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space, being N the number of features, that distinctly classifies the data points.

The Support Vector Regression (SVR) adopts identical principles as the SVM for classification, with only a few minor differences. As the name suggests, it is a regression algorithm, where the output is a real number with infinite possibilities. The main idea is the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated. More specifically:

1. Assume that the equation of the hyperplane is $y_i = x_i\beta + b$. To ensure that is as flat as possible, the objective function is formulated as a convex optimization: minimize $\frac{1}{2}\|\beta\|^2$.

   Subject to all residuals having a value less than ε, $|y_i - (x_i\beta + b)| \leq \varepsilon$;

2. However, it is possible that no such function exists to satisfy these constraints for all points. To deal with otherwise infeasible constraints, introduce slack variables ξ_i and ξ_i^* for each point. The slack variables allow regression errors to exist up to the value of ξ_i and ξ_i^*, yet still satisfy the required conditions. Including slack variables leads to the objective function:

$$minimize\ \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \qquad with \quad \xi_i \geq 0 \quad and \quad \xi_i^* \geq 0 \tag{4.2}$$

   Subject to $y_i - (x_i\beta + b) \leq \varepsilon + \xi_i$ and $(x_i\beta + b) - y_i \leq \varepsilon + \xi_i^*$.

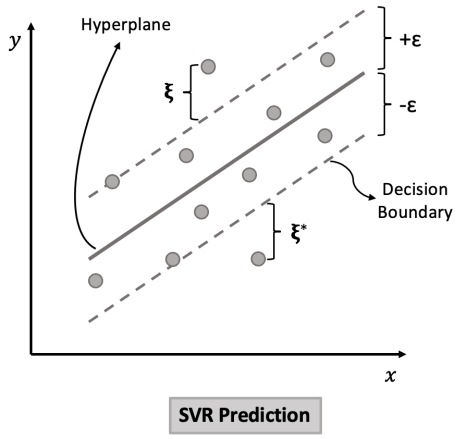

Figure 4.19: Flow chart of a Support Vector Machine Regression Algorithm

The SVR algorithm seeks to fit the error inside a threshold value as opposed to other regression models that try to reduce the error between the actual and projected value. It is simpler to implement and robust against outliers. SVR provides a proficient prediction model while acknowledging the non-linearity in the data. However, in cases where the number of features for each data point exceeds the number of training data samples, the SVR will underperform as well as when the data set has more noise. It might not be suitable for large datasets.

4.4.1.4 Neural Networks

NN, also known as Artificial Neural Networks, are computational models that are capable of extracting meaning from imprecise or complex data. The learning process finds patterns and detects trends, similar to how the human brain works, where biological neurons signal to each other. This concept was first proposed by Turing (1948).

A neural network contains an input layer, zero or more hidden layers, and an output layer. Each layer has one or more nodes (figure 4.20). Each node connects to the nodes in the next layer, and each connection has an associated weight. Starting from input nodes, a neural network produce an output, which is the solution of a problem. The workflow is briefly described:

1. Define the connection weight between nodes. A gradient descent technique is used to compute coefficients to minimize the cost function. During the iterations these values will be updated in order to reach the best predicted value.

2. Each layer's output is calculated forwardly by an activation function, f. The activation function can be linear or nonlinear.

   Commonly, a bias node (a constant, typically initialized to 1) is added to each input layer to shift the activation function.

3. The obtained value from the activation function will be the final value of this neuron at the current step, and continues until it reaches the final output.

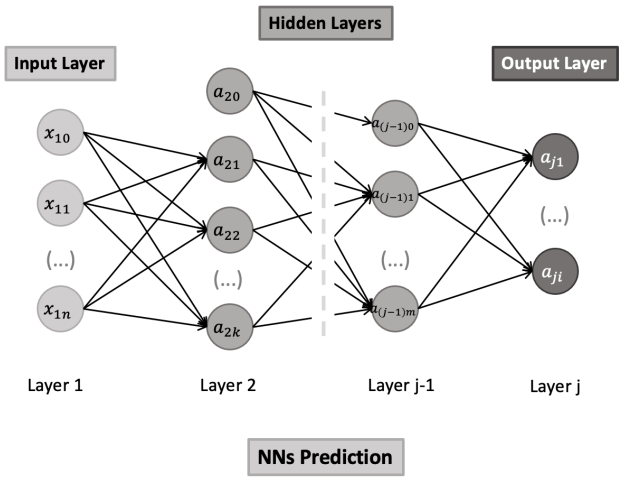

Figure 4.20: Flow chart of a Neural Network Algorithm

Neural networks are flexible and can be used for both regression and classification problems. It is reliable in an approach of tasks involving nonlinear data with a large number of inputs and many features. After training, they are able to extract from a huge continuous stream of data only the information necessary for them, ignoring all extraneous noise. Once trained, predictions are made rather quickly. However, the hardware cost increases with the complexity of the problem, and its setup requires additional effort to maintain.

The greater amount of data used during training, the more accurate the results are. Dependency on data is one of the leading disadvantages of NN, as some have to be on the maintenance side to watch it. Furthermore, most neural networks are black-box systems, generating results based on experience and not on specified programs, making it difficult for modifications.

4.4.2 Test Design

To build a reliable machine learning model, it is necessary to split the dataset into distinct sets. The training set is the set of data that is used to train and make the model learn the hidden features and patterns in the data. The validation set is used to validate model performance during training. This validation procedure gives information that may be used to adjust the hyperparameters and settings of the model. The main idea of splitting the dataset into a validation set is to avoid overfitting the model. After training is complete, the model is tested using a different set of data called the test set. It provides an unbiased final model performance metric in terms of accuracy and precision.

To help explain the test design made in this project, consider figure 4.21 that illustrates a business whose contract starts in January. As previously described, one of the insurer's tasks is to predict, at the beginning of the tenth month of contract, the claims that will still arrive in the future, referring to accidents occurred in the first 9 months. Forecasting claims that occur at the tenth, eleventh and twelfth month is another different issue. As a result, claims related to the accident months of October, November, and December are disregarded (represented in dark grey).

Regarding the remaining months, what is interesting for the insurer is to have a model with the ability to learn data that are not yet known. For instance, claims that happened in January and were reported in the same month (delay 0), in February (delay 1),..., and in September (delay 8) are already known. The same for claims that occurred in February. Those reported in the same month (delay 0) to those reported in September (with delay 7) are already known. What is not known a priori are the claims that will be reported from October on wards whose accident month was between January and September (which correspond to the yellow part). So, the ones that fall within the light grey area are removed.

The same reasoning was taken into account for the other contracts that do not start in January: considering the tenth, eleventh and twelfth months and the delays of the remaining months.

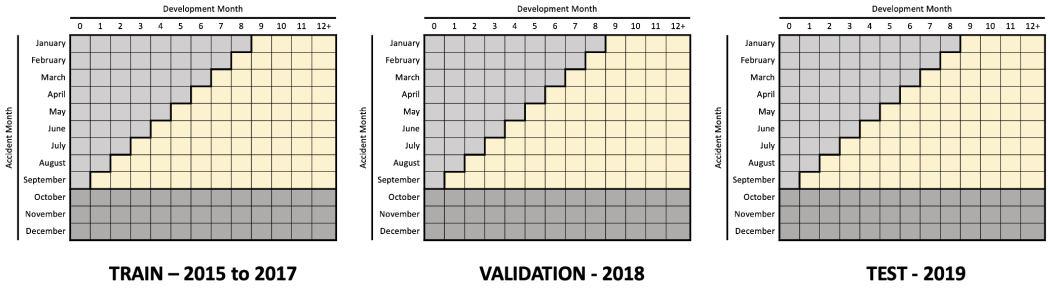

Figure 4.21: Slip representation between train, validation and test

After selecting the appropriate data to train the model, the test design follows. For the training set, were considered the contracts that start between 2015 and 2017, corresponding to 17.872 observations (55%). For the validation set, were selected contracts that started in 2018 and for the test set, contracts of 2019, each corresponding to 7.344 observations (22.5% each).

4.4.3 Data Reduction - Feature Importance

The feature importance describes which variables are relevant for use in model construction. It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, improve the performance of the model. This will aid in a better understanding of the solved problem.

For this, the Random Forest algorithm is a particularly effective tool. It computes the average impurity decrease from all decision trees in the forest to determine feature importance. There are two key measures to take into account: the mean decrease accuracy, which computes the feature importance on permuted out-of-bag samples based on the mean decrease in accuracy (the prediction error for regression is Mean Squared Error (MSE)); and the mean decrease impurity, which is the total decrease in node impurities from splitting on the variable, averaged over all trees (for regression, it is measured by the residual sum of squares). Figure 4.22 illustrates these measures when the RF algorithm is applied.

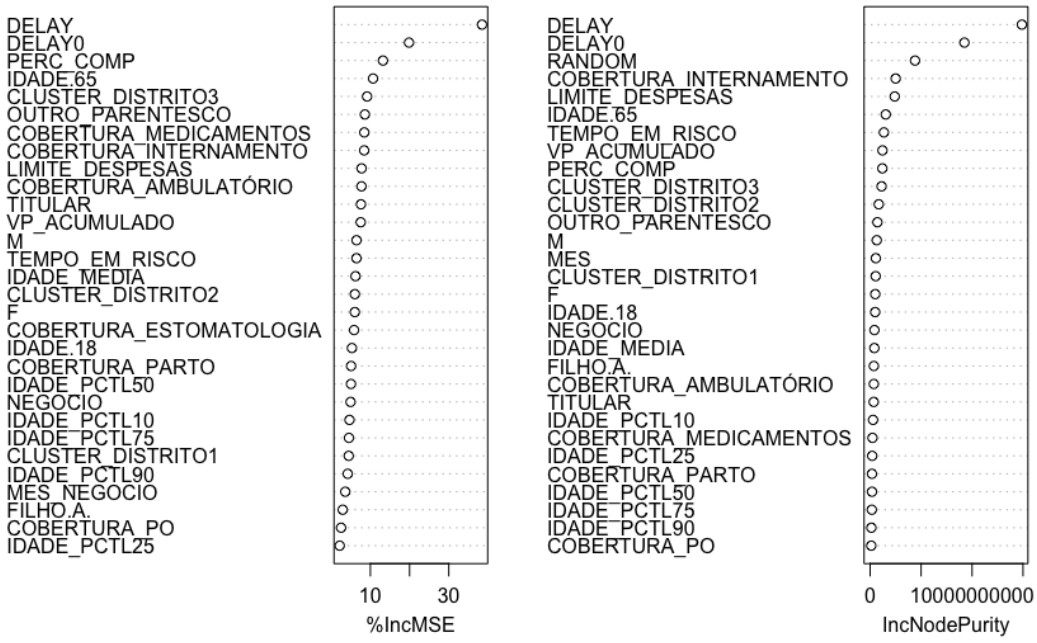

Figure 4.22: Feature importance output when using a Random Forest

The first fundamental point is how crucial the delay is to the problem (as previously seen, the longer the delay, the less likely the insurer will receive an expense). That is why having an idea of the value that arrived in the month in which the claim happened ('DELAY0') is also important. If the amount that arrives is less than expected, it may indicate that more claims will arrive in the upcoming months. There are additional variables that were chosen for the model, despite the fact that the RANDOM variable appeared as one of the first ones. The selection process was based on choosing the first variables in the left table, which did not have a correlation greater than 0,8.

The following variables were then selected to reflect the IBNR phenomenon:

- 'DELAY'
- 'DELAY0'
- 'PERC_COMP'
- 'LIMITE_DESPESAS'
- 'IDADE.65'
- 'CLUSTER_DISTRITO3'
- 'COBERTURA_MEDICAMENTOS'
- 'COBERTURA_INTERNAMENTO'
- 'COBERTURA_AMBULATÓRIO'

The variables that represent each insured person's plans are significant factors to consider - the higher the plafond, the greater the expense the insurer will have to be responsible for. The number of elderly or those residing in Portugal's interior can capture some effects (of lower costs) that the remaining variables are unable to.

There is one variable that also stands out: 'COBERTURA'. Given its importance and the fact that different coverages have different weights in the overall cost of the IBNR component, it was decided to build different models in addition to the model with all coverages: one for the Outpatient coverage, another for Inpatient (the ones that cost the insurer the most), and another for the remaining coverages. This could enhance the forecasting power of the models because each coverage represents a different phenomenon and might have different variables to better explain it. In the end, it will be determined whether the model that includes all coverages is better than the combination of the three models that consider each coverage individually.

Therefore, the dataset was divided into 3: one corresponding to Outpatient coverage; one for Inpatient coverage; and, one for all other coverages together. Then, 3 new feature importance were made, one for each submodel.

4.4.3.1 Feature Selection: Outpatient model

Given the output of the importance that Random Forest gives to Outpatient data (figure 4.23), the following variables were selected:

- 'DELAY'
- 'DELAY0'
- 'IDADE.65'
- 'OUTRO_PARENTESCO'
- 'IDADE_PCTL50'

The plafond and the percentage of reimbursement are no longer as significant because they are quite similar in all Outpatient coverage - regardless of the plan that the company chooses for its employees. For this phenomenon, the median age of each company and the number of people who are not holders are considered.

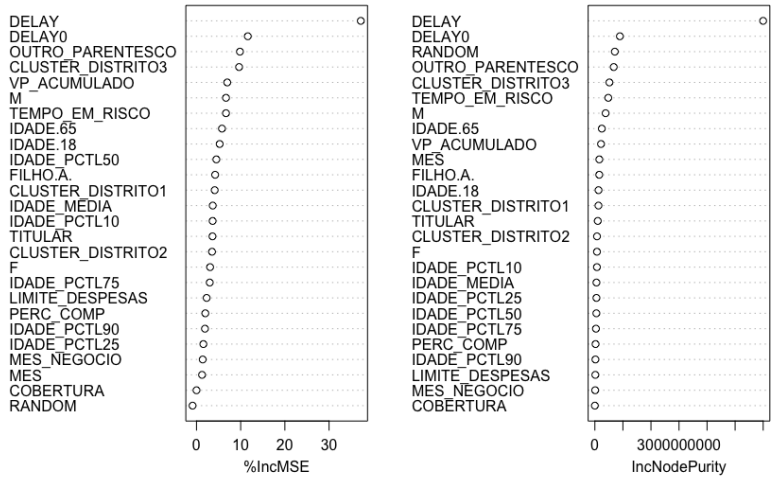

Figure 4.23: Feature importance of the Outpatient data

4.4.3.2 Feature Selection: Inpatient model

When considering Inpatient data (figure 4.24), the most important variables are the following:

- 'DELAY'
- 'DELAY0'
- 'IDADE.65'
- 'IDADE_PCTL50'
- 'TEMPO_EM_RISCO'
- 'LIMITE_DESPESAS'
- 'PERC_COMP'

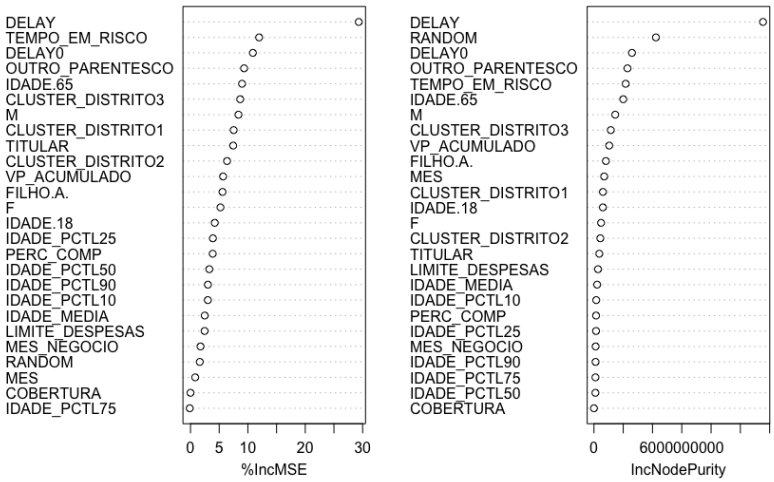

Figure 4.24: Feature importance of the Inpatient data

As seen in the previous outputs, the variables related to delay and age are still significant with regard to Inpatient coverage. Here, the number of people is also relevant since the likelihood of 1 or more people being hospitalized increases with the number of people insured by a company's health insurance (thus, the opposite is also

true: the fewer people a company has, the lower the probability of hospitalizations). Additionally, the conditions of the plan (plafond and percentage of reimbursement) are important because, depending on the company, the plan can be quite different.

4.4.3.3 Feature Selection: Remaining coverages model

And finally, the feature importance output for coverage of Stomatology, Prostheses and Orthoses, Childbirth and others (figure 4.25). The chosen variables are a mixture of what was presented:

- 'DELAY'
- 'DELAY0'
- 'PERC_COMP'
- 'LIMITE_DESPESAS'
- 'COBERTURA_PARTO'
- 'COBERTURA_PO'
- 'COBERTURA_ESTOMATOLOGIA'
- 'COBERTURA_MEDICAMENTOS'
- 'IDADE.65'
- 'CLUSTER_DISTRITO3'

Figure 4.25: Feature importance of the Remaining coverages data

This time, with the specification of each coverage belonging to this dataset, since each one represents a different phenomenon. Once again, the importance of the delay, the conditions of the plan, the number of people over 65, and the number of people living in the interior of Portugal are highlighted.

4.4.4 Build Model

Most modeling techniques have multiple parameters or settings that can be adjusted to control the modeling process. Typically, a model is first built using the default options, and then parameters are refined during subsequent trainings.

4.4.4.1 Building Random Forest model

To apply the RF algorithm, it was used the *randomForest()* function from the *randomForest* package (version 4.7). Although there aren't many options available in this package to optimize the function's parameters, it does have the benefit of running more efficiently. Even so, there is an argument that can be enhanced with *tuneRF()* - the *mtry*, the number of variables randomly sampled at each split. Usually, the default value is considered to be the number of variables divided by 3. The tunning function *tuneRF()* searches for the optimal value with respect to Out-of-Bag error estimate, so it doesn't take into account the validation set. Therefore, this model was trained with the training and validation datasets together (as well as the others, the difference is that there was no differentiation to optimize the parameter). The value chosen to train each of the 4 models is represented in table 4.6.

There is also a parameter that defines the number of trees to grow - *ntree*. Despite the fact that increasing the number of trees increases the precision of the outcome, the value of *ntree* will remain at the default value since 500 trees are adequate enough.

Table 4.6: Optimized parameter for the RF model

Parameter	All coverages	Outpatient	Inpatient	Remaining
mtry	4	2	4	4

4.4.4.2 Building Extreme Gradient Boosting model

One of the biggest advantages that makes this algorithm one of the preferred is that it is highly customizable and has some very important parameters that can be changed in each different situation. For this purpose, the *xgboost* package (version 1.6.0.1) was chosen.

To simplify hyperparameter tuning, only 2 parameters were optimized through a simple grid search: *eta* which is the step size shrinkage used in update to prevent overfitting (the default value is 0.3); and *max.depth* the number of splits in each tree (the default value is 6). These parameters were optimized and are shown in table 4.7.

The following task parameters were selected to fit this particular problem and were shared by all models: the *objective* as *reg:squarederror*; and the *eval_metric* as *mae*. This means that the loss function to be minimized is a regression with squared loss and the chosen evaluation metric for validation data is the mean absolute error. All other arguments remained at their default values.

Table 4.7: Optimized parameters for the XGBoost model

Parameter	Grid Search	All cov.	Outpatient	Inpatient	Remaining
eta	(0.01,0.02, 0.1,0.2,0.3)	0.01	0.1	0.02	0.3
max.depths	(5,6,7,8,9,10)	5	5	5	5

4.4.4.3 Building Support Vector Regressor model

The *svm()* function from the *e1071* package (version 1.7) was implemented to apply SVM models. They were trained using a linear kernel, fitted to a regression model by setting the type equal to *eps-regression*. Therefore, there is only one parameter that needs to be optimized: the cost, which represents misclassification or error term. The misclassification tells the SVM optimization how much error is bearable. When the cost is high, it will accurately classify every data point and there is a possibility of overfitting.

Additionally, as was previously demonstrated, an epsilon tolerance margin must be established to approximate the SVM. The parameters that maximized the mean absolute error in the validation set were selected as the ones to train models (table 4.8).

Table 4.8: Optimized parameters for the SVR model

Parameter	Grid Search	All cov.	Outpatient	Inpatient	Remaining
cost	(50, 75, 100, 125)	100	100	100	100
epsilon	(0.1, 0.5, 1)	0.1	1	1	0.1

4.4.4.4 Building Neural Networks model

When configuring a neural network to a dataset, it is crucial to ensure optimal data scalability. Without it, a variable's scale alone could have a significant impact on the prediction variable. Unscaled data could produce meaningless results. For scaling the data in this case, min-max normalization was used.

A neural network is one of the most complex algorithms. It is essentially a black box so it's more challenging to discuss fitting, weights, and the model itself. Therefore, and given the low computational power, it was decided to change only the number of layers and the number of hidden neurons in each layer for each model. To do this, the *neuralnet()* function from the *neuralnet* package (version 1.44.2) was used.

Table 4.9 displays the selected hidden layers. Although some layer combinations were tested, the chosen values were arbitrary. Still, some heuristics were considered: as the data has few features, the number of hidden layers should be between 1 and 3; the number of hidden neurons should be between the size of the input layer and the output layer; and the number of hidden neurons should keep on decreasing in subsequent layers.

Table 4.9: Optimized parameter for the NN model

Parameter	All coverages	Outpatient	Inpatient	Remaining
hidden layers	2 layers: 5 and 3 nodes	2 layers: 3 and 2 nodes	2 layers: 4 and 2 nodes	3 layers: 5, 3 and 2 nodes

4.4.5 Verify *Chain Ladder* assumptions

Before applying the model, it is necessary to ensure that its main assumptions are verified. In Mack's 1994 paper, a procedure was designed to test for calendar year influences (T. Mack, 1994). The *ChainLadder* package (version 0.2.15), available in R (Gesmann et al., 2022), provides some functions that simulate those procedures. This package was also used for all *Chain Ladder* method-related topics.

One of the main assumptions is the non-correlation of individual development factors. The *dfCorTest()* function uses Spearman's correlation coefficient to test this assumption. It is returned a statistic T that is assumed to be normally distributed. As a result, it is possible to provide a confidence interval threshold to evaluate the test's outcome. If the metric is within the confidence interval, therefore the development factors are not correlated (Gesmann et al., 2022).

Table 4.10 shows the results for each business and for each model to be applied. The data used in each instance are the test data from known claims (for a visual representation, see figure 4.21, test matrix, cells in light gray). If the output is 'False' then the development factors are not correlated and the *Chain Ladder* method can be applied. Otherwise, if it is 'True', the development factors are correlated, and, in theory, the method could not be applied. In cases where the output is 'NA', it indicates that at least one of the columns of the triangle under study has the value 0. This means that it is not possible to calculate the correlation, much less the *Chain Ladder* method since it would have a development factor that had been divided by 0, which is impossible.

Table 4.10: Test for proportionality between development years

NEGOCIO	All Coverages	Outpatient	Inpatient	Remaining
A	True	True	True	True
B	False	False	False	True
C	True	False	True	False
D	True	True	NA	False
E	True	True	False	False
F	True	True	NA	True
G	True	True	NA	True
H	False	True	NA	False
I	False	False	NA	False
J	NA	NA	NA	NA
K	NA	NA	NA	NA
L	NA	NA	NA	NA
M	False	True	NA	False
N	NA	False	NA	NA
O	False	NA	NA	False
P	False	NA	NA	False
Q	NA	NA	NA	NA
R	False	True	NA	False

Another basic assumption is the independence between accident years, in this case, between accident months. The *cyEffTest()* tests for it in a similar way. If the output metric is within the confidence interval, therefore the triangle doesn't have a Monthly Calendar Effect (Gesmann et al., 2022). Results are displayed in table 4.11. The above-mentioned interpretation holds.

Table 4.11: Test for independence between accident months

NEGOCIO	All Coverages	Outpatient	Inpatient	Remaining
A	False	True	False	True
B	False	True	False	True
C	False	False	False	False
D	True	True	True	True
E	False	False	False	False
F	True	False	True	False
G	False	NA	False	False
H	False	False	True	False
I	False	False	False	False
J	False	False	NA	False
K	False	False	NA	True
L	False	False	NA	True
M	False	NA	NA	False
N	False	False	NA	False
O	True	False	NA	True
P	False	False	NA	False
Q	False	False	NA	False
R	False	NA	False	False

For the data under study, the non-correlation between development factors is more challenging to ensure. Only 30% of the 72 matrices tested guaranteed this assumption. Results at the level of 'All coverages' are easier to obtain because more data is available. For the remaining models, given that the data are split up, there are fewer observations, and therefore it becomes more challenging to demonstrate the assumption. Additionally, it seems that using the *Chain Ladder* approach becomes more difficult the smaller the insured universe is (i.e., the fewer employees a company has). In general, it can be said that there is a correlation between the subsequent months.

The independence between accident months is easier to verify, being true for 62% of the matrices tested. Here, it should be noted once again that smaller businesses will have fewer claims, which means a lower likelihood of hospitalization, and thus, many missing values.

As a result, the traditional *Chain Ladder* method might not be the ideal one to apply as it cannot guarantee the assumptions it requires, particularly in smaller businesses. Although the insurer does not calculate the coverage level, this method should be applied with caution.

4.4.6 Assess Model

Performance measures are essential for supervised machine learning models to evaluate and track the performance of their predictions. Such metrics add substantial and necessary value in model selection and model assessment and can be used to evaluate different types of models. 4 error metrics are commonly used for regression models:

4.4.6.1 R-Squared (R^2)

The coefficient of determination, or R^2, is the proportion of variation in the outcome that can be explained by the predictor variables. It is a measure that provides information about the goodness of fit of a model. In the context of regression, it is statistical indicator of how closely the regression line resembles the actual data. The closer R^2 is to 1, the better the model.

$$R^2 = 1 - \frac{Sum\ of\ Squares\ of\ Residuals}{Total\ Sum\ of\ Squares} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \overline{y}_i)^2} \tag{4.3}$$

4.4.6.2 Mean Squared Error (MSE)

MSE assesses the average of the squares of the errors, that is, the average squared difference between the observed actual outcome values and the values predicted by the model. When a model has no error, the MSE equals zero. As model error increases, its value increases.

$$MSE = Mean((Observed - Predicted)^2) = \frac{\sum(y_i - \hat{y}_i)^2}{Number\ of\ observations} \tag{4.4}$$

4.4.6.3 Root Mean Squared Error (RMSE)

Mathematically, the RMSE is the square root of the MSE. So, the lower the RMSE, the better the model. The RMSE is the used to return the MSE error to the original unit by taking the square root of it while maintaining the property of penalizing higher errors. MSE is more difficult to interpret and is more sensitive to outliers in absolute terms.

$$RMSE = \sqrt{MSE} \tag{4.5}$$

4.4.6.4 Mean Absolute Error (MAE)

The MAE measures the prediction error similarly to RMSE. It is the average absolute difference between observed and predicted values. It is a linear score, which means that all the individual differences are weighted equally in the average. MAE is less sensitive to outliers compared to RMSE.

$$MAE = Mean(Abs(Observed - Predicted)) = \frac{\sum(|y_i - \hat{y}_i|)}{Number\ of\ observations} \tag{4.6}$$

Chapter 5

Results and Discussion

In this chapter, the results of each model and their respective performances are given, employing all the methodologies previously presented. The models will be tested in the test subset, which is the final contract year available for each business.

The output of each model is given at the corporate business level for each month, and delay, i.e., the value of each yellow cell in figure 4.21 is filled in. The first performance measures will be at the level of this output.

However, the ultimate goal of this project is to have the most accurate results possible for each business, regardless of the month or the delay. The insurance company wants to predict after the first 9 months of the contract, the amount of claims that will arrive referring to these 9 months. Therefore, the error at a global level will also be analyzed, that is, the error that compares the total sum of estimates per business. These results will be more significant in the profitability analysis of ML models when compared to the traditional one. Furthermore, the *ChainLadder* package only provides outputs at the monthly level without considering the delay, so it is only possible to compare the traditional method on global errors.

5.1 'All coverages' models

Table 5.1 then represents the first performance measures of the models. These are obtained directly from what comes out of the models - at the business, month, and delay level.

Table 5.1: Performance measures of 'all coverages' models

Model	R2	MAE	MSE	RMSE
RF	0,84	795	4.629.478	2.151
XGBoost	0,83	755	4.751.250	2.179
SVM	0,03	1.170	27.035.432	5.200
NN	0,33	25.025	657.303.924	25.637

Tables 5.2 and 5.15 represent the performance of the models when the outputs are summed per business.

Table 5.2: Sum of 'all coverages' outputs per business (in euros)

NEG.	Observed	CL	RF	XGBoost	SVM	NN
A	160.851	136.747	260.691	184.602	6.834	-1.784.102
B	1.756	68.368	-18.771	13.328	28.380	-2.191.478
C	109.169	102.455	139.710	94.777	41.952	-2.121.865
D	32.758	58.383	17.064	37.346	9.268	-2.209.037
E	-6.001	22.493	17.879	11.589	9.098	-1.906.516
F	-10.295	33.484	14.247	11.892	13.270	-2.219.767
G	14.067	16.101	18.179	13.879	2.379	-1.586.200
H	11.461	9.089	19.428	12.175	11.877	-1.903.943
I	1.792	21.800	6.344	7.451	7.655	-1.902.214
J	-238	-10.710	1.210	2.310	-228	-951.278
K	5.820	2.867	1.027	2.757	8.487	-1.902.952
L	4.465	9.368	5.726	4.914	7.398	-1.902.842
M	1.538	5.255	6.116	3.635	3.146	-1.587.000
N	1.902	7.056	-154	3.458	3.538	-949.897
O	548	1.588	4.362	5.052	9.876	-1.900.601
P	5.067	1.848	4.419	3.672	7.050	-1.583.795
Q	1.478	4.552	6.163	2.007	4.760	-1.586.373
R	4.177	11.729	5.327	5.901	7.717	-1.902.513

Table 5.3: Performance measures of the models when the 'all coverages' outputs are summed per business

Model	R2	MAE	MSE	RMSE
RF	0,94	14.227	718.455.285	26.804
XGBoost	0,96	6.583	100.491.345	10.024
SVM	0,14	19.720	1.699.419.961	41.224
NN	0,03	1.801.815	3.380.862.954.724	1.838.712
CL	0,77	14.546	506.622.612	22.508

The expected prediction effect could not be captured by either the SVM or the NN models. They present the poorest performance measure outcomes. MSE is excessively high and R^2 is too low, in particular for neural networks, meaning that the models did not adjust to the regression. NN generates similar outputs and far beyond the actual values. The number of layers might not be adequately adapted to the problem, or the standardization performed might not be the most correct. There are many more sophisticated experiments that could enhance NN performance, but the complexity required to apply them to real business scenarios is not worthwhile. Furthermore, the behaviour of the SVM model is unstable. Analyzing the absolute values reveals that in 5 cases (businesses E, H, J, K, and M), it turns out to be the one that manages to predict better. In other instances, though, it ends up failing so miserably that its performance metrics end up being inadequate. It is not a suitable choice because of its instability.

The tree-based algorithms, on the other hand, had quite satisfactory results. At the individual level, both RF and XGBoost had very similar results. But when summed up by business, there is an obvious highlight for XGBoost, which has lower MAE and RMSE. In fact, in the 3 cases where RF was better at forecasting (businesses I, P, and R), XGBoost had similar predictions. The mean absolute error of this model is around 6.500 euros.

To this day, *Chain Ladder* calculations are made without taking coverage into account. Therefore, the results presented in this section are those that correspond to what is currently being done. On average, the insurer is missing around 14.500 euros per business. Surprisingly, despite how simple the calculations are to perform, it turns out to be a really effective method. However, it only outperforms the ML models in 2 of the 18 businesses (C and O). When comparing only the CL method and the best ML model (XGBoost), it is possible to see that the new model made better predictions for 15 businesses and has better performance measures.

5.2 Single models

As previously explained, given the importance of the coverage variable, a comparison was also carried out by separating the data into Outpatient, Inpatient, and the remaining ones. The results are presented in the following subsections. Given that many of the behaviours are repeated and the result of aggregating these results will be the most important, there won't be very extensive analysis in this section.

5.2.1 'Outpatient' models

When looking at only the most representative coverage - Outpatient - the results are as follows:

Table 5.4: Performance measures of 'Outpatient' models

Model	R2	MAE	MSE	RMSE
RF	0,65	145	540.761	735
XGBoost	0,69	136	379.594	616
SVM	0,03	828	1.763.675	1.328
NN	0,61	715	895.542	946

If only the performance measures are compared, an enormous recovery of the NN model is observed, both from the R^2 and the RMSE. However, table 5.5 shows that the absolute results per business are well below expectations. Just like for the SVM model.

The outcomes for the remaining models are fairly good, with XGBoost showing a clear highlight, being closer to the observed values in 12 of the clients. Even though it has also a good performance, the CL model is consistently surpassed by the XGBoost.

Table 5.5: Sum of 'outpatient' outputs per business (in euros)

NEG.	Observed	CL	RF	XGBoost	SVM	NN
A	43.757	111.461	52.798	36.031	30.540	-4.965
B	5.535	16.057	18.048	13.729	-494	-34.435
C	8.308	50.417	36.125	22.519	-247	-30.981
D	12.915	23.237	13.840	9.206	-14.399	-44.007
E	-1.835	5.871	7.541	3.149	3054	-41.807
F	-1.439	8.015	5.485	3.691	-12.645	-45.885
G	4.520	6.436	10.031	4.530	-2.157	-43.407
H	2.055	3.539	4.853	1.901	-19.410	-46.714
I	2.096	5.478	3.918	1.837	-10.310	-46.098
J	1.435	2.199	1.960	1.439	-17.552	-46.470
K	1.093	294	1.514	661	-22.312	-46.612
L	1.034	1.461	1.582	1.163	-17.819	-45.995
M	420	1.594	1.404	1.154	-22.891	-46.310
N	1.343	1.173	1.427	759	-28.760	-45.632
O	133	557	544	493	-34.231	-46.133
P	851	1.192	920	573	-29.111	-46.128
Q	264	1.012	692	555	-31.619	-46.120
R	426	2.846	1.624	1.154	-21.366	-46.264

Table 5.6: Performance measures of the models when the 'Outpatient' outputs are summed per business

Model	R2	MAE	MSE	RMSE
RF	0,77	4.522	66.326.009	8.144
XGBoost	0,8	2.662	21.984.042	4.689
SVM	0,56	19.134	451.245.371	21.243
NN	0,86	46.493	2.176.559.996	46.654
CL	0,91	8.993	374.993.588	19.365

5.2.2 'Inpatient' models

Looking at the results of the inpatient coverage data:

Table 5.7: Performance measures of 'Inpatient' models

Model	R2	MAE	MSE	RMSE
RF	0,57	758	4.559.629	2.135
XGBoost	0,58	707	4.482.422	2.117
SVM	0,01	1.233	10.730.014	3.276
NN	0,47	4.768	26.569.080	5.155

Hospitalizations are a phenomenon that is considerably harder to predict, and the results reflect this. The R^2 score falls abruptly in all cases.

Nevertheless, as shown in table 5.8, the conventional CL model would be incapable of predicting this coverage on its own for clients with lower claims amounts. This occurs because the CL cannot forecast during the months when the claims are null (don't

Table 5.8: Sum of 'Inpatient' outputs per business (in euros)

NEG.	Observed	CL	RF	XGBoost	SVM	NN
A	52.447	-70.585	198.216	160.855	-25.739	-286.005
B	-23.229	21.684	-83.067	-34.182	-75.497	-373.593
C	58.913	-10.293	26.919	9.723	-10.225	-312.062
D	5.423	-642	-12.948	-4.114	-20.725	-317.162
E	-4.512	7.261	1.385	3.858	6.000	-312.402
F	-12.409	16.536	1.627	-774	-31.201	-320.090
G	338	744	-905	-457	-8.492	-316.929
H	2.784	-507	9.792	1.182	-18.086	-300.377
I	-5.252	NA	-6.082	-1.487	-26.356	-320.253
J	-1.673	NA	-1.013	-1.186	-12.963	-320.786
K	2.023	NA	1.776	-1.347	-21.698	-321.394
L	731	NA	1.074	-405	-28.800	-322.625
M	320	NA	2.048	-57	-22.720	-321.252
N	-1.194	NA	-2.601	664	-12.625	-321.766
O	-895	NA	938	-381	-11.687	-321.921
P	3.632	NA	1.001	266	-19.029	-321.765
Q	533	NA	1.535	-593	-20.073	-322.045
R	282	NA	-710	-1.014	-26.654	-321.895

Table 5.9: Performance measures of the models when the 'Inpatient' outputs are summed per business

Model	R2	MAE	MSE	RMSE
RF	0,57	16.435	1.471.731.668	38.363
XGBoost	0,48	12.099	813.140.116	28.516
SVM	0,1	26.992	1 096.408.261	33.112
NN	0,4	324.032	105.233.511.099	324.397
CL	0,62	35.954	2.870.983.361	53.582

exist). For that reason, it is possible to determine that selecting the RF or XGBoost models would be a wise decision for 10 of the clients. For the remaining 8 clients, who present claims amounts in all months, the CL only proves to be better in 2 cases.

5.2.3 'Remaining coverages' models

When adding data from Stomatology, Prostheses and Orthoses, Medicines, Childbirth, and other coverages the results are as follows:

Table 5.10: Performance measures of 'remaining coverages' models

Model	R2	MAE	MSE	RMSE
RF	0,69	61	138.636	372
XGBoost	0,64	65	154.580	393
SVM	0,02	92	416.235	645
NN	0,39	453	528.103	727

Table 5.11: Sum of 'remaining coverages' outputs per business (in euros)

NEG.	Observed	CL	RF	XGBoost	SVM	NN
A	64.646	111.500	59.705	57.350	5.135	-72.127
B	19.450	26.953	44.228	37.570	23.309	-43.300
C	41.948	60.016	51.389	45.332	11.715	-27.423
D	14.420	27.154	12.709	9.932	399	-140.126
E	346	7.532	3.245	5.571	1.944	-98.051
F	3.553	8.484	6.476	9.717	1.717	-132.252
G	9.210	9.835	7.420	6.781	534	-85.582
H	6.622	6.658	8.766	6.307	3.780	-108.085
I	4.948	11.209	5.641	8.227	3.487	-113.276
J	0	NA	686	343	-804	-28.880
K	2.704	2.035	2.469	2.297	4.350	-111.992
L	2.700	6.568	3.161	3.044	3.995	-112.112
M	7.99	3.724	2.093	2.421	1.953	-84.123
N	1.753	NA	956	821	2.122	-28.017
O	1.310	NA	2.645	2.197	2.687	-109.839
P	584	NA	1.226	738	3.660	-82.822
Q	680	NA	2.148	1.837	2.013	-83.350
R	3.468	NA	4.205	4.123	3.767	-112.401

Table 5.12: Performance measures of the models when the 'remaining coverages' outputs are summed per business

Model	R2	MAE	MSE	RMSE
RF	0,89	3.276	42.419.912	6.513
XGBoost	0,91	3.178	27.871.255	5.279
SVM	0,16	7.522	265.456.459	16.293
NN	0,09	97.383	10.591.827.383	102.917
CL	0,98	9.305	239.978.373	15.491

Again, the CL proved to be incapable of making some predictions. Despite its good R^2 score, it only reveals forecasts closer to those observed in 2 clients (businesses G and H). On the other hand, the SVM and NN models continue to show the same behavior, however, it should be noted that for businesses B, E, F, M, N, and R the SVM forecasts are closer to the real. Anyway, in this scenario, XGBoost would also be the chosen model.

5.3 Summed single models: 'Outpatient', 'Inpatient' and 'Remaining'

Results are presented in a similar way to the previous ones. Here, a new perspective is given to the insurer: making calculations at the coverage level in order to try to obtain some particular effect. Table 5.13 shows, at the business, month, and delay level, what it would be like if the insurer had models that summed up forecasts for Outpatient,

Inpatient, and other coverages individually.

Table 5.13: Performance measures of summed single models

Model	R2	MAE	MSE	RMSE
RF	0,82	817	4.899.051	2.213
XGBoost	0,83	779	4.810.907	2.193
SVM	0,03	2.102	28.067.883	5.298
NN	0,73	6.788	51.172.823	7.154

Tables 5.14 and 5.15 represent the performance of the models when those outputs are summed per business. As it is possible to observe in table 5.2, the same observed values are being compared under 2 different perspectives.

Table 5.14: Sum of 'single' outputs per business (in euros)

NEG.	Observed	CL	RF	XGBoost	SVM	NN
A	160.851	152.376	310.719	254.236	9.935	-363.097
B	1.756	64.695	-20.791	17.117	-52.682	-451.328
C	109.169	100.140	114.433	77.575	1.243	-370.466
D	32.758	49.749	13.602	15.024	-34.725	-501.295
E	-6.001	20.664	12.171	12.579	10.998	-452.259
F	-10.295	33.035	13.588	12.635	-42.129	-498.227
G	14.067	17.014	16.546	10.853	-10.115	-445.918
H	11.461	9.690	23.412	9.390	-33.716	-455.176
I	1.792	16.686	3.477	8.577	-33.179	-479.627
J	-238	2.199	1.633	595	-31.319	-396.135
K	5.820	2.329	5.758	1.610	-39.659	-479.998
L	4.465	8.029	5.818	3.802	-42.624	-480.731
M	1.538	5.317	5.546	3.518	-43.658	-451.686
N	1.902	1.173	-218	2.244	-39.264	-395.415
O	548	557	4.127	2.309	-43.231	-477.893
P	5.067	1.192	3.146	1.576	-44.480	-450.715
Q	1.478	1.012	4.375	1.799	-49.679	-451.516
R	4.177	2.846	5.119	4.263	-44.252	-480.560

Table 5.15: Performance measures of the 'single' models when the outputs are summed per business

Model	R2	MAE	MSE	RMSE
RF	0,90	15.209	1.359.320.594	36.869
XGBoost	0,86	12.519	624.408.028	24.988
SVM	0,43	52.047	3.634.169.306	60.284
NN	0,45	467.909	220.091.629.717	469.139
CL	0,82	11.484	404.895.718	20.122

Despite a significant improvement in their performances, the SVM and NN models remain unable to predict desirable outcomes. Also, both tree-based algorithms lose performance when making predictions this way. When comparing XGBoost, for most clients, the perspective that calculates all coverages together turns out to be better (except for J, M, N, O, Q and R businesses). So the mean absolute error rises from 6.500 to 12.500 euros.

The *Chain Ladder* turns out to have better performance measures from this perspective, as evidenced by an increase in R^2 from 0.77 to 0.82. In fact, it is forecasting better for companies with fewer claims. What happened in these cases is that it was predicting much more what actually occurred. It is now predicting slightly lower values, which helps in forecasting this type of business. Businesses with more claims may find that their forecast has also improved, which demonstrates a better ability to adapt to any behaviour. The mean absolute error goes from 14.500 to 11.400 euros. The forecast using the *Chain Ladder* technique improved for 13 of the 18 clients (excluding C, G, K, M, and P businesses). The insurance company was predicting a total of around 162.000 euros above the observed; after this change, it would predict 148.000 euros above, which represents an improvement of 14.000 euros.

Chapter 6

Conclusions

This project compares the traditional *Chain Ladder* model with machine learning algorithms when predicting claim reserves. For this purpose, a database of corporate clients of a health insurance company was used. The main objective was to identify the most accurate way to determine these provisions.

As ML models, the performances of a RF algorithm, an XGBoost algorithm, a SVM algorithm for regression, and a simple NN algorithm were tested. In addition to comparing the 4 new models with the *Chain Ladder* technique, it was also provided to the insurer a new perspective on performing these calculations by considering Outpatient, Inpatient, and remaining coverages separately. Each of these predictions attempted to capture any distinctive behaviour that one of these coverages might exhibit.

The neural network produced the worst results, with extremely high MAE and RMSE and very low R^2. It was unable to accurately capture the behaviour of the regression under study. Its application might be too complex for real-world business scenarios, so it is not worthwhile to conduct a more thorough investigation on the matter. The SVM model, on the other hand, proved to be unstable, with some values quite accurate and others quite distant from the actual value, which also proved to be inadequate.

Tree-based algorithms had very satisfactory results, with the XGBoost outperforming slightly better the RF model. When forecasting all coverages together, XGBoost has a mean absolute error of 6.500 euros, while RF has 14.200 euros. Its results make this the best ML algorithm for predicting IBNR claims. In neither of the two, forecasting by coverage has improved the results.

As also seen in the literature, the *Chain Ladder* approach offers admirable results considering how straightforward the computations are, with an R^2 score of 0.77. On average per business, this current methodology has an absolute error of 14.500 euros for the IBNR estimates. If the insurance company started to calculate the *Chain Ladder* by coverage, it would achieve a significant improvement in its forecasts, reducing the mean absolute error per client to 11.400 euros.

This method's worth has been demonstrated once more, and its popularity over the years has justified it. However, the insurer neglects to consider the fact that, in theory, it does not verify the assumptions that it requires, as can be observed. The majority of the research data does not ensure proportionality between the development months, and there is no independence between accident months. On the other hand, if a business is smaller, that is, if it has fewer claims and thus many null values, the model becomes unable to generate forecasts because it is unable to calculate one or more development factors.

A way to overcome these obstacles is by using more sophisticated algorithms that are able to adjust to the complexity needed in each situation, as is the case with the XGBoost algorithm. It is a resilient and reliable approach that prevents excessive overfitting quite easily, with output predictions that are easy to handle. However, it must be taken into account that the algorithm is still a black box. In real-world business scenarios, there is sometimes a need to understand and explain to non-specialized co-workers and clients how calculations are made. It is therefore more difficult for these models to gain acceptance within the business. Even for those who work with this kind of algorithms, the overall method is hardly scalable. This is because the estimators base their accuracy on prior predictors, making it difficult to streamline the process.

Still, compared to the current method, the XGBoost model that forecasts all coverages together compensates significantly. For the analysed clients, the *Chain Ladder* would predict around 162.000 euros above the observed, while XGBoost predicted only 80.400 euros above, which represents a gain of 81.600 euros and an improvement of 50%. The mean absolute error per client has improved by around 8.000 euros.

So, the most accurate method to predict IBNR claims is by using an XGBoost algorithm that takes into account the delay to predict, health insurance conditions (plafond and co-payment percentage), number of people with more than 65 years old, district of living, coverages and the claims amount where the accident month is the same as the reporting month. Thus, predictions will be better than they would be using the traditional approach.

Chapter 7

Limitations and recommendations for future works

The project's major limitation was definitely the computational power available, which severely constrained the project's scope. Several attempts were made to improve the models, but the tools available made the process very time-consuming and were eventually left out of the study. For instance, the *caret* package would have been an ideal package to implement ML models and tuning parameters in R. However, it requires cross-validation techniques and a dataset of 20.000 rows proved to be quite heavy for this purpose (failed attempt after running for 8 hours). This was unfortunate given that one of the biggest requirements in this project is that it would not make sense to make a random division between training, validation, and testing (that's why it ended up being split into 2015 to 2017, 2018, and 2019 respectively) and the *caret* package allowed a manual choice for this division. There was even an attempt to switch to Python, to explore if it was feasible to find more efficient packages that satisfied the same requirement. It was discovered and tried to implement the *hypopt* package that was specifically designed to manually define the design between training and validation. Still, it was also demanding a lot of computing power.

Working with such a large amount of data using these resources was a really difficult challenge. Given the growth of data that is been observed, it would be interesting to start considering working with servers as an alternative to local processors.

Despite being aware of the potential for improvement, it was decided to run the models mostly using default parameters. Even with those default parameters, it should be noted that quite satisfactory results were achieved. However, computational power also limited the sample under study. The insurance company in question has a huge portfolio of corporate clients and only 18 were analyzed. One of the recommendations for future work would be to study the application of the models to different samples of clients and determine whether the performance is maintained or not. Thus, it is possible to more confidently apply these IBNR projections across the entire portfolio.

Furthermore, as analyzed in the results, the traditional *Chain Ladder* method offers the best predictions for clients with fewer claims, especially when calculations are made at the coverage level. It would be interesting to study a hybrid model in which computations are performed using XGBoost for the largest clients and *Chain Ladder* for the smaller ones. For this, it would be necessary to find an optimal threshold that categorizes clients as large or small based on the number of claims they submit annually. Finally analyzing whether this new perspective could be even better than having just one model tailored to any case.

It would also be crucial to investigate whether the results are consistent with more current data to understand how the COVID-19 virus has impacted the behaviour and forecast of these provisions. Therefore, consider the accident years of 2020, 2021 and 2022.

References

Bornhuetter, R. L., & Ferguson, R. E. (1972). The actuary and ibnr. In *Proceedings of the casualty actuarial society* (Vol. 59, pp. 181–195). (Cit. on pp. 1, 9).

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140. doi:https://doi.org/10.1007/BF00058655. (Cit. on p. 31)

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. doi:10.1023/A:1010950718922. (Cit. on p. 31)

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T. P., Shearer, C., & Wirth, R. (2000). Crisp-dm 1.0: Step-by-step data mining guide. (Cit. on p. 13).

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). doi:10.1145/2939672.2939785. (Cit. on p. 32)

Duval, F., & Pigeon, M. (2019). Individual loss reserving using a gradient boosting-based approach. doi:10.3390/risks7030079. (Cit. on p. 11)

England, P., & Verrall, R. (2002). Stochastic claims reserving in general insurance. *British Actuarial Journal*, *8*(3), 443–518. doi:10.1017/S1357321700003809. (Cit. on p. 9)

Gesmann, M., Murphy, D., Zhang, Y. (, Carrato, A., Wuthrich, M., Concina, F., & Dal Moro, E. (2022). *Chainladder: Statistical methods and models for claims reserving in general insurance*. R package version 0.2.15. Retrieved from https://CRAN.R-project.org/package=ChainLadder. (Cit. on pp. 44, 45)

John, A. (2018). Granular reserving dialogistic in machine learning. Institute and Faculty of Actuaries. (Cit. on p. 10).

Kotsalo, N. (2021). *Machine learning methods vs. traditional methods in forecasting loss reserves*. Master Thesis. Arcada University of Applied Sciences. Retrieved from https://urn.fi/URN:NBN:fi:amk-2021060213705. (Cit. on p. 11)

Lourenço, J. M. (2021). *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. Retrieved from https://github.com/joaomlourenco/novathesis/raw/master/template.pdf. (Cit. on p. iii)

Mack, T. (1994). Measuring the variability of chain ladder reserve estimates. *Insurance: Mathematics and Economics*, *1*, 101–182. doi:https://doi.org/10.1016/0167-6687(94)90721-8. (Cit. on p. 44)

Mack, T. C. (1993). Distribution-free calculation of the standard error of chain ladder reserve estimates. *ASTIN Bulletin*, *23*, 213–225. (Cit. on pp. 6, 7, 9).

Mack, T. C. (1999). Measuring the variability of chain ladder reserve estimates. (pp. 101–182). (Cit. on p. 9).

McGuire, G., Taylor, G., & Miller, H. (2018). Self-assembling insurance claim models using regularized regression and machine learning. *51*. doi:10.2139/ssrn.3241906. (Cit. on p. 10)

Mulquiney, P. (2006). Artificial neural networks in insurance loss reserving. In *Proceedings of the 9th joint international conference on information sciences (jcis-06)*. doi:10.2991/jcis.2006.67. (Cit. on p. 11)

Skurnick, D. (1973). A survey of loss reserving methods. *Proceedings of the Casualty Actuarial Society*, *60*, 16–58. (Cit. on p. 1).

Turing, A. (1948). Intelligent machinery. doi:10.1093/oso/9780198250791.003.0016. (Cit. on p. 35)

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc. (Cit. on p. 34).

Wang, Z., Wu, X., & Qiu, C. (2021). The impacts of individual information on loss reserving. *ASTIN Bulletin*, *51*(1), 303–347. doi:10.1017/asb.2020.42. (Cit. on p. 10)

Wirth, R., & Hipp, J. (2000). Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the fourth international conference on the practical application of knowledge discovery and data mining* (pp. 29–39). (Cit. on p. 13).