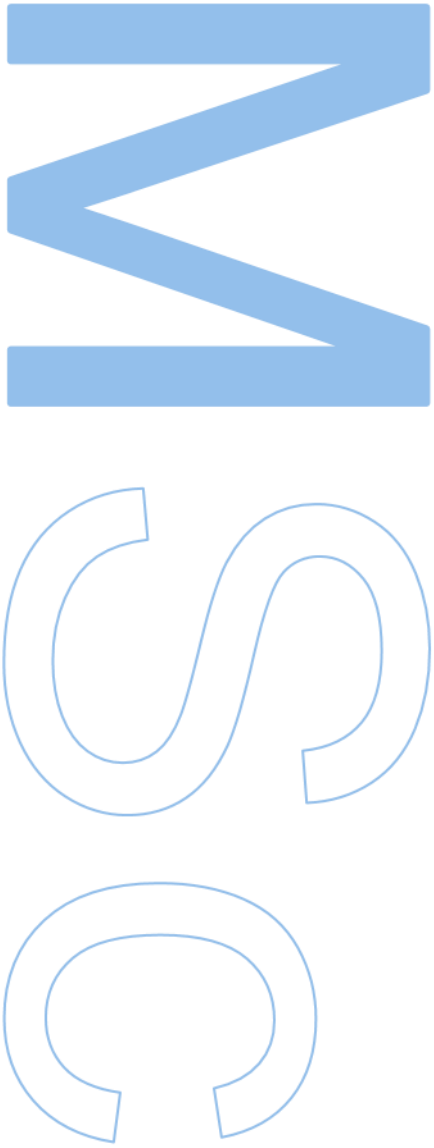
A Machine Learning Approach for Predicting Claims Reserving

Amanda Custódio Tavares
Mestrado em Ciência de Dados
Departamento de Ciência de Computadores
2023

Orientador
Rita Paula Almeida Ribeiro, Professora Auxiliar, Faculdade de Ciências da Universidade do porto



" In God we trust. All others must bring data "

Edwards Deming

Acknowledgements

I would like to express my gratitude for all the support given to me during this time. I want to thank specially my supervisors, Rita P. Ribeiro, from University of Porto, Luis Maranhão and Luis Filipe Ferreira from Ageas, they all gave contribution from different perspectives and different expertise. Also, dedicated their time, knowlodge and patience to help me everytime I needed.

To my boyfriend, Philippe, who always believed in me even when I did not.

To my friends Arina and Lirielly, who walked this path with me and were always very supportive.

To my family, that always encouraged and were there for me even having some of them living in a different continent.

Abstract

Insurance companies offer coverage against financial losses resulting from unexpected events such as accidents, illnesses, and property damage. Our case study focused on motor insurance claims from 2019 to 2022 from a given insurance company. The goal was to determine whether a machine learning approach can give an effective solution for predicting claims reserving and which variables help explaining those predictions.

Given this purpose, our first objective was to build a regression model to estimate the ultimate claim cost. We used ensemble learning decision tree algorithms such as Random Forest, Gradient Boosting, and Extreme Gradient Boosting (XGBoost). These algorithms are known for their effectiveness in several domains. However, understanding the reasoning behind predictions is crucial, particularly in the insurance sector, where decisions involve costs and thus require justification. Therefore, for the second objective, we used post-hoc explanation methods such as feature importance from decision tree ensembles and Shapley Values (SHAP), a well-known method in the area of eXplainable Artificial Intelligence.

Our experimental study showed that Gradient Boosting consistently yielded slightly better results in terms of RMSE. However, when accounting for asymmetric loss functions - which are relevant for this domain, XGBoost was the best from the Huber Loss point of view, and Random Forest excelled in terms of Lin Lin Loss. This implies that the choice of the algorithm may depend on which error metric we aim to minimize.

The interpretability analysis with SHAP and feature importance helped us understand the model's estimates. It became evident that certain features, such as premium, played a significant role in determining claim cost predictions.

Overall, our results suggest that while some features were revealed to be irrelevant, providing too much detail, such as car brand groups and the driver's zone of residency, having more informative features that characterize the degree of damage would be extremely important.

From this perspective, our study is an important contribution to the insurance company, as few projects leverage machine learning methods in the financial sector.

Keywords: Claims Reserving, Motor Insurance, Machine Learning, Chain Ladder, Predictive Modeling, Regression Model.

Resumo

Companhias de seguros oferecem cobertura contra perdas financeiras resultantes de eventos inesperados, como acidentes, doenças e danos materiais. O nosso estudo de caso concentrou-se em sinistros de seguro automóvel de 2019 a 2022 de uma determinada companhia de seguros. O objetivo é determinar se uma abordagem baseada em Aprendizagem de Máquina é efetiva para a estimativa das reservas e quais variáveis mais influenciam essas previsões.

Com base neste propósito, o nosso primeiro objetivo foi construir um modelo de regressão para estimar o custo final do sinistro.

Utilizamos algoritmos baseados em árvores de decisão em conjunto, como *Random Forest*, *Gradient Boosting* e *Extreme Gradient Boosting (XGBoost)*. Esses algoritmos são conhecidos pela sua eficácia em várias áreas. No entanto, compreender o raciocínio por trás das previsões é crucial, especialmente no setor segurador, onde as decisões envolvem custos e, portanto, exigem justificação. Portanto, para o segundo objetivo, utilizamos métodos de explicação pós-processamento, como a importância das variáveis das árvores de decisão combinadas e os valores de Shapley (SHAP), um método bem conhecido na área da Inteligência Artificial Explicável.

O nosso estudo experimental mostrou que o *Gradient Boosting* consistentemente proporcionava resultados ligeiramente melhores em termos de RMSE. No entanto, ao considerar funções de perda assimétricas - que são relevantes para este domínio, o XGBoost foi o melhor do ponto de vista da Função de Huber, e o *Random Forest* se destacou em termos da Função de Lin Lin. Isso implica que a escolha do algoritmo pode depender da métrica de erro que pretendemos minimizar.

A análise de interpretabilidade com SHAP e a importância das características ajudaram-nos a compreender as estimativas do modelo. Tornou-se evidente que certas características, como o prémio, desempenharam um papel significativo na determinação das previsões de custo do sinistro.

No geral, os nossos resultados sugerem que, embora algumas características tenham se revelado irrelevantes, fornecer muitos detalhes, como grupos de marcas de automóveis e a zona de residência do condutor, ter mais características informativas que caracterizem o grau de dano seria extremamente importante.

Deste ponto de vista, o nosso estudo é uma contribuição importante para a companhia de seguros, pois poucos projetos aproveitam métodos de aprendizado de máquina no setor financeiro.

Palavras-chave: Reservas de Sinistros, Seguro Automóvel, Aprendizagem de Máquina, Chain Ladder, Modelagem Preditiva, Modelo de Regressão.

Contents

Acknowledgements iii

Abstract v

Resumo vii

Contents viii

List of Figures xi

List of Tables xiii

Glossary xv

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Objectives . . . 2
- 1.3 Organization . . . 3

2 Background 5
- 2.1 Actuarial Background . . . 5
- 2.2 Machine Learning . . . 7
  - 2.2.1 Ensemble Algorithms . . . 8
  - 2.2.2 Shapley Additive Explanatins (SHAP) . . . 10
- 2.3 Machine Learning Approaches in the Actuarial Business . . . 11

3 Case Study on Claims Reserving 15
- 3.1 Data Understanding and Preprocessing . . . 15
- 3.2 Claims Dataset . . . 24

4 Experimental Study 27
- 4.1 Experimental Setup . . . 27
  - 4.1.1 Feature Selection . . . 28
  - 4.1.2 Hyperparameter Tuning . . . 31
- 4.2 Performance Estimation . . . 33
- 4.3 Interpretability Analysis . . . 39

4.4 Discussion 42

5 Conclusions **43**
5.1 Main Contributions 43
5.2 Limitations and Future Work 44

A Features Description **47**

Bibliography **49**

List of Figures

2.1 Simplified generic ensemble architecture [5] 8

3.1 Original and transformed target distribution. 20
3.2 Spearman's correlation matrix. 21
3.3 Quantitative features distribution. 22
3.4 Cost boxplot of categorical features. 23

4.1 Baseline model's feature importance - Random Forest. 34
4.2 Real cost vs expected cost in log scale - Gradient Boosting - Baseline. 34
4.3 Histograms for real and expected cost in log scale - Gradient Boosting - Baseline. 35
4.4 SHAP's summary barplot - Gradient Boosting - Baseline. 40
4.5 SHAP's beeswarm plot - Gradient Boosting - Baseline. 40
4.6 SHAP's waterfall plot - highest error Gradient Boosting - Baseline. 41
4.7 SHAP's force plot - highest error Gradient Boosting - Baseline. 41

List of Tables

2.1 Claims development triangle 7

3.1 Features types and description by group 19
3.2 Claims per year during the period 2019-2022 20
3.3 Features in claims dataset 25

4.1 Feature selection results 30
4.2 RMSE training results from model without hyperparameter tuning 31
4.3 Hyperparameter search spaces 32
4.4 RMSE results - Training set - Baseline 33
4.5 RMSE results - Test set - Baseline 33
4.6 Range results - Gradient Boosting - Baseline 35
4.7 Scenarios descriptions 36
4.8 RMSE results for 10-Fold Cross Validation training 37
4.9 Error metrics during Test 39

A.1 Features - Claims Dataset 47
A.2 Features - Policy Dataset 48
A.3 Features - Join Data Dataset 48

Glossary

CANN	Combined Actuarial Neural Network
ccODP	cross-classified Over-Dispersed
CL	Chain Ladder
GLM	General Linear Model
IBNER	Incurred But Not Enough Reported
IBNR	Incurred But Not Reported
LDFs	Loss Development Factors
LoBs	Line of Business
MAE	Mean Absolute Error
MDN	Mixture Density Networks
RBNS	Reported But Not Settled
RFE	Recursive Feature Elimination
RMSE	Root Mean Squared Error
SHAP	Shapley Additive Explanations
XGBoost	Extreme Gradient Boosting

Chapter 1

Introduction

Insurance companies have long been dealing with the problem of claims reserving. There are different parametric and non-parametric methods to deal with it, and they typically use aggregated claims data based on their occurrence and payment dates. With the development and popularization of Machine Learning models, the idea is to explore a different approach to this problem. In this chapter, we will present the main motivations of this dissertation, its objectives, and how the document is structured.

1.1 Motivation

Insurance companies are businesses that provide coverage against financial losses due to unforeseen events such as accidents, illnesses, and property damage. They do this by pooling the risk of policyholders, meaning that many people pay into a fund, and then those who experience a covered loss are paid from that fund, usually a wide range of products is offered.

Insurance companies are heavily regulated by government agencies to protect policyholders and ensure that they are financially stable. They must meet certain standards and maintain certain reserve levels to ensure they can pay claims. It is also required to file financial statements and other information with regulatory bodies, which allows them to be monitored and evaluated by the authorities.

Claims reserving is a process used to estimate the amount of money that will be needed to pay for claims that have been made but have not yet been settled. The process involves making assumptions about the likelihood and cost of future claims and setting aside money in a reserve account to cover those costs. The goal of claims reserving is to

ensure that an insurance company has enough money on hand to pay all of its claims, even if they are more expensive or more numerous than expected. The liabilities are a very important part of measuring the company's solvency, while underestimation affects the resources available to fulfill the liabilities, overestimation affects the profitability.

Nonetheless, there exist various scenarios in which a time gap emerges between the factual occurrence date of an event and the point at which it is communicated to the insurance provider and factored into the financial statement. Such instances may arise due to claims being documented after a certain delay, extended periods for processing claims settlements, reopening of claim proceedings, or limited availability of comprehensive claim details [1].

The increasing demand is evident for the careful choice of suitable reserving techniques and assumptions, aiming for practicality as much as precision, especially when dealing with data that may not be flawless. It must be taken into account the duration of the insurance agreement, the nature of the coverage offered, and the probability of a claim arising. Furthermore, insurers need to adapt their computations to accommodate changing circumstances.

This work will focus on motor insurance, more specifically on material damage, divided between own damage and third-party liability.

1.2 Objectives

The main objective of this study is to take advantage of the high amount of data available throughout the company and explore new methodologies. In order to do that, the goal is to build a regression model based on specific claims features instead of aggregated data. This study aims to understand if the model brings benefits when compared to the traditional methodology that has been used.

To accomplish this goal, different techniques must be used to understand the data, its vulnerabilities, and its potential. The dataset available has real data from Grupo Ageas Portugal from 2019 until 2022. It contains data from the insurance policy, the insured vehicle, and the usual driver. The first approach to the dataset was to analyze the information about closed claims, only for light motor vehicles related to material damage with own damage coverage.

1.3 Organization

This chapter started by introducing the motivation and goals for this project, and then a structure could be expected as follows. In Chapter 2, we present a literature review, starting with the problem statement to contextualize the problem, and then we go through other similar studies in the field, where different machine learning approaches were tested in the actuarial context.

During Chapter 3, a context of insurance companies and an actuarial background with Mack's Chain Ladder, a very spread methodology used in claims reserving, are presented. Following that, we have the exploratory data analysis to help understand the data, the variables, and the data preprocessing.

In Chapter 4, the regression model is built after choosing the best features using feature selection methods, hyperparameter tuning, and exploring possible scenarios. Then, the results are discussed and SHAP is used to help explain the results.

Lastly, there is Chapter 5 with the conclusions that can be taken from this study, with its main contributions, limitations, and future work.

Chapter 2

Background

This chapter explores the existing literature on claims reserves involving traditional and Machine Learning approaches. The examination of previous academic works aims to explore the methodologies, models, and variables that impact the determination of claims reserves. This effort seeks to make a meaningful contribution to advancing this crucial aspect of the insurance sector. As we work on this project, our main objective is to give the insurance company helpful information that makes it easier to manage their reserves wisely.

2.1 Actuarial Background

At its core, insurance provides policyholders with a sense of financial security. Premiums are collected from policyholders in exchange for the insurance company's commitment to offer compensation in the event of covered losses. Reserves are established to maintain this commitment and ensure the company's long-term financial health.

Maintaining reserves cannot be overstated within the insurance industry, a domain full of uncertainty. Usually, insurance companies have a diverse array of coverage types, which is no different here. However, throughout our analysis, our focus narrows to a specific subset: own damage coverage of motor insurance.

The loss reserving provision has always been part of the insurance companies, it is very important in the company's report to calculate the liabilities, and investment allocation, and also it influences pricing decisions. It allows the company to cover all possible outstanding loss liabilities.

It is even more important and challenging when it comes to non-life insurance line of business (LoBs) because the claims usually have many payments until it is closed and might be reported with a long delay for several reasons like court decisions or further investigation. The time gap between the occurrence and reporting dates is called reporting delay.

Accidents that have been reported to the insurer but have not been resolved yet are termed as reported but not settled (RBNS). On the other hand, accidents that have taken place but are still unknown to the insurer are referred to as incurred but not reported (IBNR) claims.

The actuary responsible for reserving is concerned with estimating the ultimate claims cost associated with accidents that have already taken place, after estimating this amount, it is possible to split the cost between payments, RBNS, and IBNR.

There are different methods to calculate it based on past aggregated data, the most common methodologies use aggregated triangular data and statistical structures such as Mack's Chain Ladder [2].

The Chain Ladder method relies on the calculation of Loss Development Factors (LDFs) to estimate reserves. These factors represent the development of claims over successive periods. The LDFs, denoted as $\hat{f}_j$, for $j = 0, 1, \ldots, n-1$, where n is the total number of development periods, are calculated as follows:

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

Using these LDFs, we derive projection factors, denoted as $\hat{F}_k$ for $k = 0, 1, \ldots, n-1$, as follows:

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

With the last known accumulated amounts and estimated development coefficients, we can estimate the values to fill the bottom triangle of the claims development table, denoted as $\hat{C}_{i,j+1}$ for $i + j > n$, $i = 0, \ldots, n$, and $j = 0, \ldots, n-1$, using the following equation:

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

Subsequently, we can calculate reserves per year of origin, denoted as $\hat{R}_i$ for $i = 0, \ldots, n$, using the following equation:

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

Finally, the total reserve $\hat{R}$, which represents the total expected value of liabilities for outstanding claims, is calculated as the sum of reserves per accident year:

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

To illustrate the Chain Ladder method with an example using a complete claims development triangle, we can consider Table 2.1, where we have fictitious numbers.

TABLE 2.1: Claims development triangle

Accident	Development Year			
Year	1	2	3	4
2019	50 000	18 000	2 000	500
2020	45 000	15 000	800	434
2021	46 000	20 000	1 411	482
2022	55 000	20 682	1 618	552

The LDFs are calculated from the paid upper triangle, in white, and then used to estimate the values shadowed in gray. Those values represent the payments since it is an incremental triangle. From this information, we can have the ultimate cost and calculate the reserves.

2.2 Machine Learning

Machine learning can provide insights into structures and patterns within large datasets. It also creates models by learning from existing datasets to predict or forecast outcomes or behavior [3].

The success of machine learning is about using the right features to build the right models that achieve the right tasks. These tasks include binary and multi-class classification, regression, clustering, and descriptive modeling. Models for the first few of these tasks are learned in a supervised fashion requiring labeled training data [4].

Unsupervised learning deals with unlabelled data, making it distinct from supervised learning, which relies on labeled data for evaluation. In unsupervised learning, tasks include clustering data and assessing the quality of such clusters by measuring the average distance from cluster centers. It can also discover associations between elements that often co-occur and identify hidden variables like film genres. Overfitting remains a concern in unsupervised learning, as excessively fine-grained clustering can lead to uninformative results.

Regarding model output, two main categories exist: predictive models that provide outputs related to the target variable and descriptive models that reveal underlying patterns in the data. Predictive models are typically learned in supervised settings, while descriptive models are generated through unsupervised methods. However, there are exceptions, such as supervised learning of descriptive models (e.g., subgroup discovery) and unsupervised learning of predictive models (e.g., predictive clustering) [4].

This study will focus on supervised learning, more specifically, a regression problem. Three ensemble methods were chosen for this task: Random Forest, Gradient Boosting, and XGBoost.

2.2.1 Ensemble Algorithms

Ensemble learning is a comprehensive meta-strategy in machine learning that aims to improve predictive performance by consolidating predictions from multiple models. This kind of algorithm is good because individual errors are compensated by the existence of other models, being stable at the same time [5]. Figure 2.1 helps illustrating how it works.

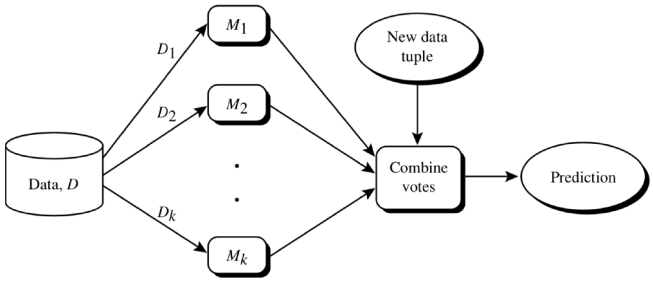


FIGURE 2.1: Simplified generic ensemble architecture [5].

Random Forest is an ensemble learning algorithm that combines tree predictors, developed by Breiman [6]. Random forests consist of multiple tree predictors, where each

tree's decisions depend on a randomly sampled vector. This vector is independently selected and follows the same distribution for all trees within the forest. As the number of trees in the forest grows, the generalization error converges almost surely to a limit. [6] It operates by constructing a multitude of decision trees during the training phase. Each tree is created using a random subset of the training data and a subset of features, introducing variability and diversity. When making predictions, these individual trees collectively contribute their outputs, and the final prediction is determined by averaging these tree predictions, considering this project is about a regression task. Random Forest addresses various limitations associated with single decision/regression trees, such as overfitting and sensitivity to noise in the data. By aggregating the outcomes of multiple trees, it enhances generalization performance and robustness. Additionally, Random Forest provides a measure of feature importance, aiding in the interpretation of the model's behavior and insights into which features have the most influence on predictions. This algorithm's adaptability to different data types, resistance to overfitting, and capacity to handle complex relationships make it a valuable choice for diverse applications. Although it is a very good algorithm, there are weak points, this algorithm is not able to extrapolate based on the training data, therefore it is not able to discover new trends.

XGBoost is an advanced ensemble learning algorithm that builds upon the foundation of decision trees, developed by Chen and Guestrin [7]. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements a parallel and distributed computing approach, combining the strengths of multiple tree predictors to achieve robust and high-performance predictions. Similar to Random Forest, XGBoost also involves constructing an ensemble of decision trees during training. However, XGBoost employs a boosting strategy, iteratively improving the model by adding trees that correct the errors of previous trees. XGBoost's key features include regularization techniques, handling missing values, and automatic handling of categorical features. This algorithm excels in both classification and regression tasks and is known for its impressive predictive power and flexibility. It is also known for its exceptional predictive performance, ability to capture complex relationships, and robustness to handle different types of data. Its capability to handle missing values and the flexibility to customize its behavior makes it a popular choice across various domains. The computational resources are very optimized when it comes to large datasets, it is an excellent option since it includes parallelization, distributed computing, and cache optimization.

Gradient Boosting is an ensemble learning technique that iteratively combines weak learners, usually decision trees, to create a strong predictive model. This approach was first introduced by Friedman [8]. Gradient Boosting constructs an additive model in a forward stage-wise manner. It starts with an initial prediction and then sequentially adds new weak learners, adjusting their parameters to minimize the residuals of the previous stage. The final model is a weighted sum of these weak learners, forming a robust and accurate predictive ensemble. This method effectively addresses shortcomings of individual decision trees by improving predictive accuracy through boosting. Each new weak learner focuses on the errors made by the previous ones, thus creating a cumulative improvement in predictive performance. Gradient Boosting's flexibility and ability to capture complex relationships make it a powerful tool for various Machine Learning tasks. It can handle both regression and classification tasks effectively. While Gradient Boosting offers high predictive accuracy, it may also face limitations, such as the risk of overfitting if not properly regularized and the potential for longer training times due to its iterative nature. Like Random Forest and XGBoost, Gradient Boosting is also constrained by the range of the training data, which can limit its ability to extrapolate beyond observed trends.

2.2.2 Shapley Additive Explanatins (SHAP)

SHAP (SHapley Additive exPlanations) by [9] is a method based on the optimal Shapley values for explaining individual predictions.

The main purpose of SHAP is to clarify the prediction generated for a given instance x by assessing each feature's contribution to the prediction outcome. SHAP employs Shapley values derived from coalitional game theory for its explanation methodology [9].

The feature values within a data instance collectively act as participants in a coalition. Shapley values provide insights into the equitable distribution of the "payout," which corresponds to the prediction, among the features. A participant can represent an isolated feature value, as seen in tabular data, or a cluster of feature values [10].

A significant innovation introduced by SHAP is the representation of Shapley value explanations as an additive feature attribution technique, reminiscent of a linear model. This perspective establishes a linkage between LIME (Local Interpretable Model-Agnostic Explanations) and Shapley values [10].

The SHAP framework outlines the explanation as follows:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

Where: - g represents the explanation model.

- z' is an element of $\{0,1\}^M$, indicating the coalition vector.

- M is the maximum coalition size.

- ϕ_j is an element of $\mathbb{R}$, signifying the feature attribution for feature j, i.e., the Shapley values.

The term "coalition vector" corresponds to "simplified features". This nomenclature seems to have been chosen due to the aggregation of image data into superpixels instead of representing them at the pixel level. Conceptually, it is helpful to consider the z 's as representing coalitions: Within the coalition vector, a value of 1 indicates the presence of the corresponding feature value, while 0 signifies its absence. This concept is reminiscent of the principles underlying Shapley values. To compute Shapley values, simulations are conducted where certain feature values are treated as "present" while others are treated as "absent." The linear model representation of coalitions serves as a technique to compute the ϕ 's. For the instance of interest x, the coalition vector x' consists of all 1's, implying that all feature values are "present." This simplifies the formula to:

$$g(x') = \phi_0 + \sum_{j=1}^{M} \phi_j$$

Shapley values are distinctive in their ability to satisfy the properties of Efficiency, Symmetry, Dummy, and Additivity [10].

2.3 Machine Learning Approaches in the Actuarial Business

One of the most common methods for estimating loss reserving is still Chain Ladder since it is distribution-free, which means it does not assume almost anything about the data, which makes it suitable for any database [11]. But this flexibility comes with a weakness, it is a very sensitive method to variation in the data during most recent accident years. There are different approaches to calculate the standard deviation of the reserve, one of them uses gamma distribution and maximum likelihood estimation [2]. Since the popularity of machine learning methods has increased and computational resources have improved, new approaches are being tested and studied.

In Aleandri [12], different approaches like Naïve Bayes, K-Nearest Neighbors, CARTs, and Neural Networks to estimate closing delay, payment amount, and case reserves for individual claims of a specific automobile bodily injury claim dataset. It was only considered RBNS claims in this paper and there was not a general conclusion since some methods had good results while others did not. The author points out constraints with data availability and initial assumptions to evaluate the methods. Also refers that it would be important to estimate IBNR reserves to be a complete work, but it was not possible because of the information available, information related to the policyholder and economic environment would be very important for this kind of analysis. The results obtained from machine learning approaches did not always outperform the traditional methods.

Sharing the same objective, Jamal et al. [13] in the report compared traditional methods with Machine Learning approaches, by showing the strengths of each one, it uses a dataset with both transactional detail and information about the policies and the insurers, it also estimates both separately IBNR and IBNER. The indicators to compare the models are yearly cashflows, total outstanding, and individual cell errors. In general, the machine learning models used did not outperform the traditional ones, but there is space to improve the models and try different techniques. It is suggested to use mixed models, and ensembles, also to consider the time dependency as future work, to improve the models used it is also suggested to combine models to incorporate both paid and incurred claim data. Like in [12], there was no policy data available therefore it was taken only a simple approach to estimate IBNR, which might be improved.

The authors Duval and Pigeon [14] also present an individual loss reserving approach based on Gradient Boosting, using information about the claims and the insured, the strategy is to estimate the ultimate claim amount. It used a dataset with 20 covariates of a Property and Casualty insurance company and different XGBoost models. The estimates generated by the XGBoost models were compared with the stochastic methods. The author also stated that this kind of model was chosen because of its strong performance for prediction on structured data and its ability to deal with the data without a lot of preprocessing. The results showed two important issues about these approaches, the estimating process could have a relevant impact because of the bias generated by the censored nature of the data, also the instability of a micro-level model based on generalized linear models, which could be avoided by combining with a macro-level approach. Despite these warnings, there were promising results achieved using XGBoost.

In [15], the author started by using Random Forest with a modified version of the cascading method like it was done by champion et al. [16] including the approach proposed by [12] which separates the predicting procedure into different parts to eliminate the limitation of the cascading method. The dataset used is from the National Association of Insurance Commissioners (NAIC) and has nine accident years and ten development years. The author compared Chain Ladder, Over Dispersed Poisson Mack Model, Neural Networks (with cascading), and Random Forest (with modified cascading). It was concluded that traditional models are still very competitive and sometimes they are better, sometimes worse.

Gabrielli [17] studied through five papers different approaches to estimate claims reserving using Neural Networks. It used Neural Networks to estimate RBNS claims reserves of six lines of business from a synthetic dataset. The results were very promising since even after changing and testing different parameters it still predominantly stayed within 5% of the true outstanding payments, also, on average, the neural network was able to correctly process the features specifics of the claim. This work leaves space to study the method to estimate also IBNRS when the policy data is available.

The Mixture Density Network(MDN) model is a Network which focuses on probabilistic forecasting by fitting a Mixed Gaussian to Incremental Claims data. This model was used in Al-Mudafer [18], which outperformed the cross-classified Over-Dispersed Poisson (ccODP) model [19] several times in terms of the central estimate, distributional and quantile estimate accuracy when applied to different datasets. The author tested other methods like General Linear Models (GLM), a combined model with GLM and MDN, and ResMDN, which is an adaptation of the Combined Actuarial Neural Network (CANN) architecture to avoid neural networks and its issues about not being easy to justify the results. The author mentions that machine learning in loss reserving literature shows that neural networks have been providing good results over GLM and Chain Ladder methods. But also, there are gaps left behind by these methods, highlighting the focus on the best estimate instead of the distribution and volatility of the values and the lack of interpretability of the Neural Networks.

In [20] the focus is on the probabilistic forecasts obtained, what is a step to fix the gap left by the Neural Network, and using Individual Claims instead of Loss Triangle. To increase their acceptance, Wüthrich and Merz [20] developed the Combined Actuarial Neural Network (CANN), which embedded a GLM into the Neural Network architecture.

This hybrid model has been applied successfully by Gabrielli et al. [21]; H. L. Poon [22] and Gabrielli [23]

An approach more focused on IBNR applied to health insurance claims was presented Sousa [24], where it compared regression models such as Random Forest, XGBoost, Support Vector Machine, and Neural Networks with the Chain Ladder methodology by creating triangles and models for each of the clients available in a specific sample.

Chapter 3

Case Study on Claims Reserving

This chapter focuses on a case study within the insurance industry, specifically examining claims from own damage coverage of motor insurance. During this chapter, we go through exploratory data analysis, data preprocessing, and feature transformations to present the case and build the necessary analysis in order to build an accurate regression model.

3.1 Data Understanding and Preprocessing

The main goal is to predict the ultimate cost claim by claim considering relevant characteristics from the policyholder and the claim, in order to do that, machine learning techniques are being tested and a regression model is built. Then, the reserves can be extracted from the difference between the ultimate claim cost and the paid amount.

Three datasets were made available: one containing aggregated information about the claim, along with its final cost up to the present moment; if it is closed, it represents the actual cost. Another dataset holds policy-related data, which can be associated with the claim. After joining the claims dataset with policy-holder information, considering only own damage claims, there are 198 743 observations (closed and open claims) and 126 003 of them are from closed claims.

Lastly, a dataset contains information about each claim payment categorized by payment type. Both the dataset with detailed information and the one with aggregated costs

include data related to both own damage claims and third-party liability claims. To capture the overall reality of the claims rather than the specifics of each individual transaction, exploratory data analysis was conducted based on the dataset containing aggregated claim information.

It is also important to keep date information, such as accident dates and accounting dates in the dataset because it will be important to compare with Chain Ladder method. The table with all features is in Appendix A.

It is very important to take advantage of feature engineering since it is common that databases do not have all the important data in a format that is meaningful to train Machine Learning models. Also, in some cases, it is necessary to transform categorical data into a numerical representation. The feature accident_year was introduced, which corresponds to the year of the incident based on the feature pdtsin. The features closing_dt and closing_year were also created, representing the closure date of the claim and its respective year based on the features Dee and Dre.

It was also necessary to remove policies from the dataset that did not reference a claimant but rather a dummy feature to keep specific accounting transactions. These policies are characterized by having "9999" starting from the seventh position of the policy number. Consequently, these cases were removed from both the claims database and the policy database.

In the policy database, there is a feature called codbonus which refers to the customer's bonus or penalty based on their claims history, known as bonus-malus. As is usual in the insurance industry, this feature ranges from -4 to 26, where values up to 7 indicate a penalty for the customer. A value of 7 signifies neither a penalty nor a bonus, while values above 7 indicate a bonus. This feature was transformed into codbonus2, where all values below 7 become -1, values equal to 7 become 0, and values above 7 become 1.

The feature protocol, when filled, indicates that the customer is associated with a protocol within Ageas Seguros, such as being part of a professional association, as, for example, a member of the Medical Association. When it's not filled (NA), it means the customer does not have any protocol. This feature has been transformed into the protocol_flag feature, which has binary values. It's set to 1 when the customer has a protocol and 0 when they don't.

The policy-holder dataset includes the birth date, driving license date, and gender of both the policyholder and the main driver. A feature called birth_dt was created, containing the birth date of the main driver when available, and the birth date of the policyholder otherwise. Similarly, the license_dt and gender features were created, following the same logic but using the driving license date and gender.

There were 69 cases where the birth date was "01/01/1850" and "01/01/1900" due to a system error. The correct birth dates couldn't be obtained, so those observations were removed from the dataset. Additionally, 17 observations were identified where the driver had more than 15 years of driving experience but was less than 25 years old. These observations were also excluded.

After these changes, a merge was performed between the claims table and the policies table, resulting in the creation of new features. The policy_duration feature represents the difference in years between the claim acceptance and the policy start date. The license_time feature represents the difference in years between the claim acceptance and the driving license date. driver_age corresponds to the driver's age at the time of the claim, calculated as the difference between claim acceptance and the birth date. car_age is the difference in years between claim acceptance and the car's construction date. The gender feature was transformed using One Hot Encoding.

We examined situations in which the manufacturing date of the automobile was later than the date of the insurance claim, suggesting a possible change in the insured vehicle since the date of the claim. Since this information is not static, we cannot definitively ascertain whether it was already updated when the claim occurred. However, it is highly improbable that this was the case for a significant number of claims

As the database reflects current information rather than information at the time of the claim, these observations were removed from the dataset. Additionally, 316 observations were excluded where the driver's age was less than 18 or greater than 100, and 23 observations were excluded where the driving license time was less than 0 or greater than 100 due to system inconsistencies.

As stated before, only own damage claims are being analyzed. Hence, the cap_od feature should be filled with the insured capital. However, sometimes, this feature is not filled because the customer asked the insurance company to cancel the own damage coverage, resulting in the cost value being entered as the own damage capital, because as

been said before the database reflects current information rather than information at the time of the claim.

The feature eng_disp reflects the car's engine displacement. However, in the case of electric vehicles, they do not have cylinders, resulting in NA values for these instances. These NAs were replaced with 0.

The feature formacob indicates the payment method chosen by the policyholder. A new feature, formacob2, was created to distinguish between payments made through bank transfers (BNC) and other methods (OUT).

The feature cob0 pertains to clients who opted for a higher value than the one that is mandatory in the third-party liability coverage which can reach a maximum of 50 million euros. It ranges from 0 to 4, representing the number of these capital requests. However, the number of observations with multiple coverages was very low. Thus, a new feature cob0_2 was created, categorizing clients who didn't choose this option as 0, and those with at least one coverage as 1.

For the features capdp, premium, eng_disp, driver_age, license_time, car_age, and policy_duration, new features were created to represent their respective divisions into bins.

Car brands were also grouped to reduce the number of categories. These groups include European, Premium, American, Korean, Japanese, and Others.

There are 21 features displayed in Table 3.1 used during this project and it can be divided into four groups of type of information: driver's characteristics, policy-related data, car's characteristics, and claims date [25]. Table 3.1 summarizes the remaining information, after all the data cleaning and data engineering. The features were chosen according to business relevance and potential insights.

TABLE 3.1: Features types and description by group

Group	Feature	Description	Type
Driver	driver_age	Age	Continuous
	gender	Gender	Categorical
	license_time	License's Time	Continuous
	region	Residential Area	Categorical
Policy Data	policy_duration	Policy Duration	Continuous
	formacob	Billing Method	Categorical
	cap_od	Own Damage Coverage Capital	Categorical
	cap_tpl	Third-Party Liability Coverage Capital	Categorical*
	ins_deductible	Insurance Deductible	Continuous
	premium	Insurance Premium	Continuous
	cob0_2	Extra Coverage 50M	Categorical
	cobonus2	Bonus-Malus	Categorical
	protocol_flag	Protocol	Categorical
Car	car_brand_group	Car Brands Group	Categorical
	eng_disp	Car Engine Displacement	Continuous
	fuel	Fuel Type	Categorical
	car_age	Car's Age	Continuous
	num_lugares2	Number of Seats	Categorical
Claims Dates	accident_year	Accident Date	Categorical
	closing_year	Closing Date	Categorical
	rep_delay	Reporting Delay	Continuous

The feature cap_tpl is annotated with an asterisk (*) in 3.1 because, despite being continuous, it assumes only three distinct values, exhibiting categorical characteristics.

During this phase of exploratory data analysis, only closed own damage claims from the dataset were utilized, where the claim cost represents the final expense. This approach aims to avoid making decisions based on individual transactions, focusing instead on the actual total claim cost. The exploratory data analysis was performed using R and Python.

This step helps to identify discrepancies and system errors within the dataset and ensure the coherence of all the information present in the dataset.

Table 3.2 depicts the total number of claims available in the dataset, and the closed claims, specifically related to closed own damage coverage claims.

TABLE 3.2: Claims per year during the period 2019-2022

Year	Total Claims	Closed Claims	Average Cost
2019	56 514	37 362	1 200
2020	45 026	29 833	1 300
2021	49 106	32 357	1 250
2022	48 097	26 451	1 400

By looking to Figure 3.1, illustrating the target distribution, is easy to notice that the cost distribution is very left skewed, nothing similar to the normal distribution, which is not the easiest scenario for a regression model to learn from. Additionally, the third quartile, at 1 452,17, confirms that despite the wide range of values, the majority of claims exhibit low costs [26]. To make the task easier for regression models, we decided to transform the target into a log scale.

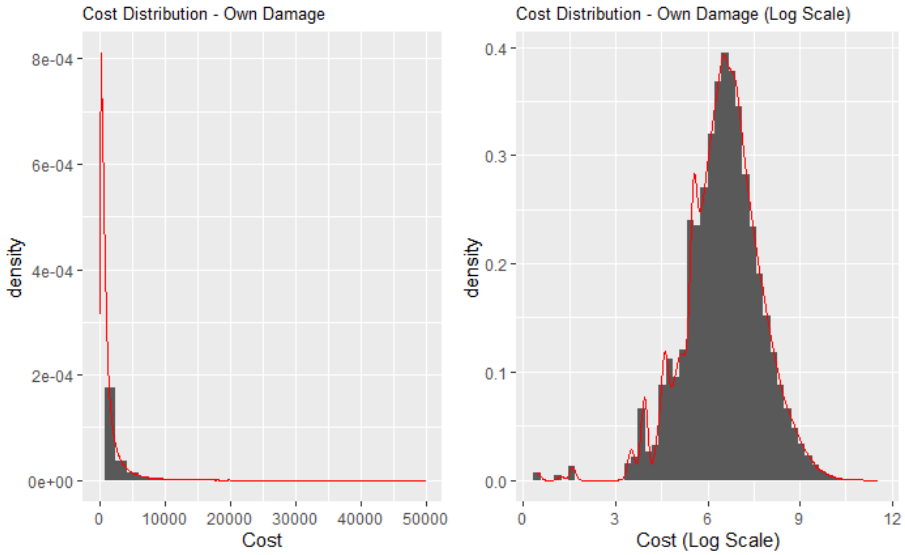


FIGURE 3.1: Original and transformed target distribution.

The correlation matrix is a very good tool to identify redundancy between features, even before using feature selection techniques might be possible to drop features if it has a high correlation coefficient, it is not necessary to keep two highly correlated features.

Only numerical features were considered to plot the correlation matrix. Figure 3.2 shows the correlation matrix obtained with the Spearman Rank Correlation Coefficient [27], which is able to capture monotonic correlations.

After examining Figure 3.2, it becomes evident that two features, namely driver_age and license_time, display a high correlation, as expected, with a correlation coefficient of 0.88. Additionally, it is apparent that there is a correlation between car_age and cap_od, although not as strong as the previous pair. This is reasonable, considering that newer cars tend to have higher own damage capital.

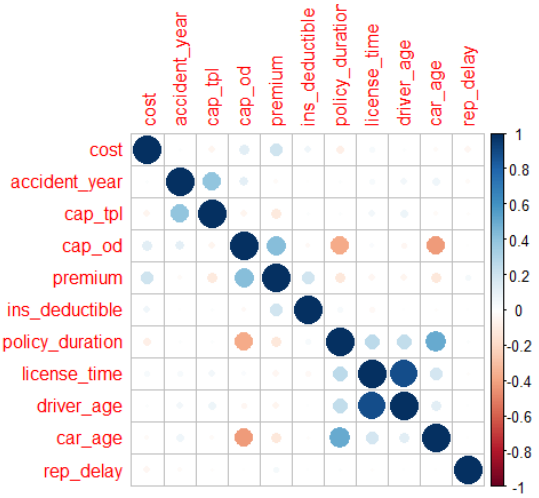


FIGURE 3.2: Spearman's correlation matrix.

To further investigate the numerical features, Figure 3.3 presents histograms illustrating the distributions of several quantitative variables. In particular, none of these distributions closely resemble a Normal Distribution. In the case of rep_delay, a prominent peak at zero suggests rapid claim reporting, indicating efficiency in this aspect.

The distribution of the premium feature exhibits a left-skewed pattern, which, although less evident, is also observable in the cap_od feature. Examining the eng_disp histogram, we detect a pronounced peak at zero, representing the electric cars. The car_age histogram displays a declining trend, reflecting the specific coverage type under analysis. Also, it suggests trends in the engine displacement.

A maximum age limit of 10 years exists for subscribing to own damage coverage. Beyond this threshold, only vehicles that had previously enrolled in the coverage are eligible to retain it.

The histograms of driver_age and license_time reinforce the correlation coefficient presented in 3.2.

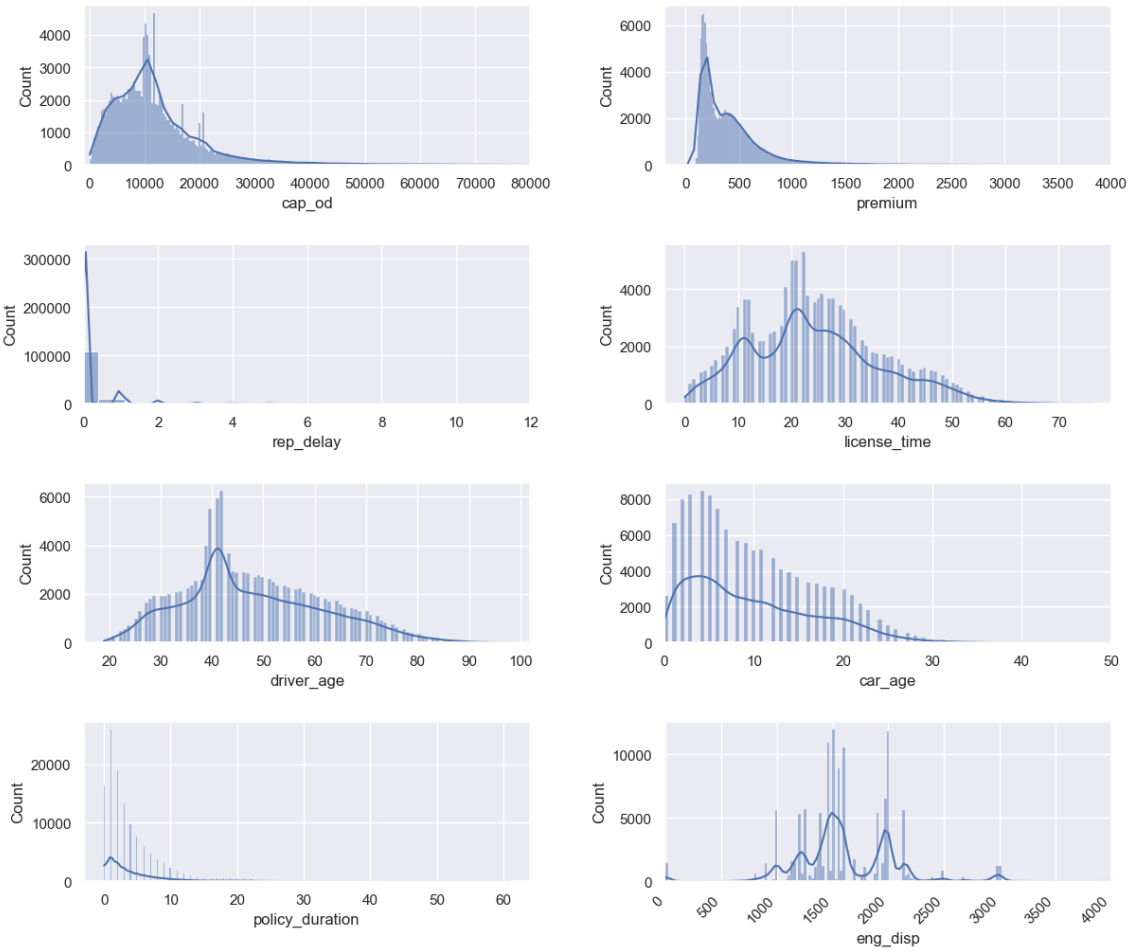


FIGURE 3.3: Quantitative features distribution.

All of the quantitative features were transformed to categorical, and divided into bins in order to see how it affects the cost, but as no significant pattern was observed it was decided to use the features as continuous as it should be better to model to be trained. Also, the categorical features were transformed with One Hot Encoding since some methodologies do not accept categorical features. These changes will be reflected in the Claims dataset.

Figure 3.4 gives an overview of the qualitative features. The car_brand feature has 66 different brands and 21 of them only have less than 10 observations. Therefore, it felt necessary to decrease the number of categories and increase the number of observations in each, which led to creating the feature car_brand_group. Now, there are only 5 categories remaining. It is a very important feature since it really affects the cost especially when it comes to own damage coverage.

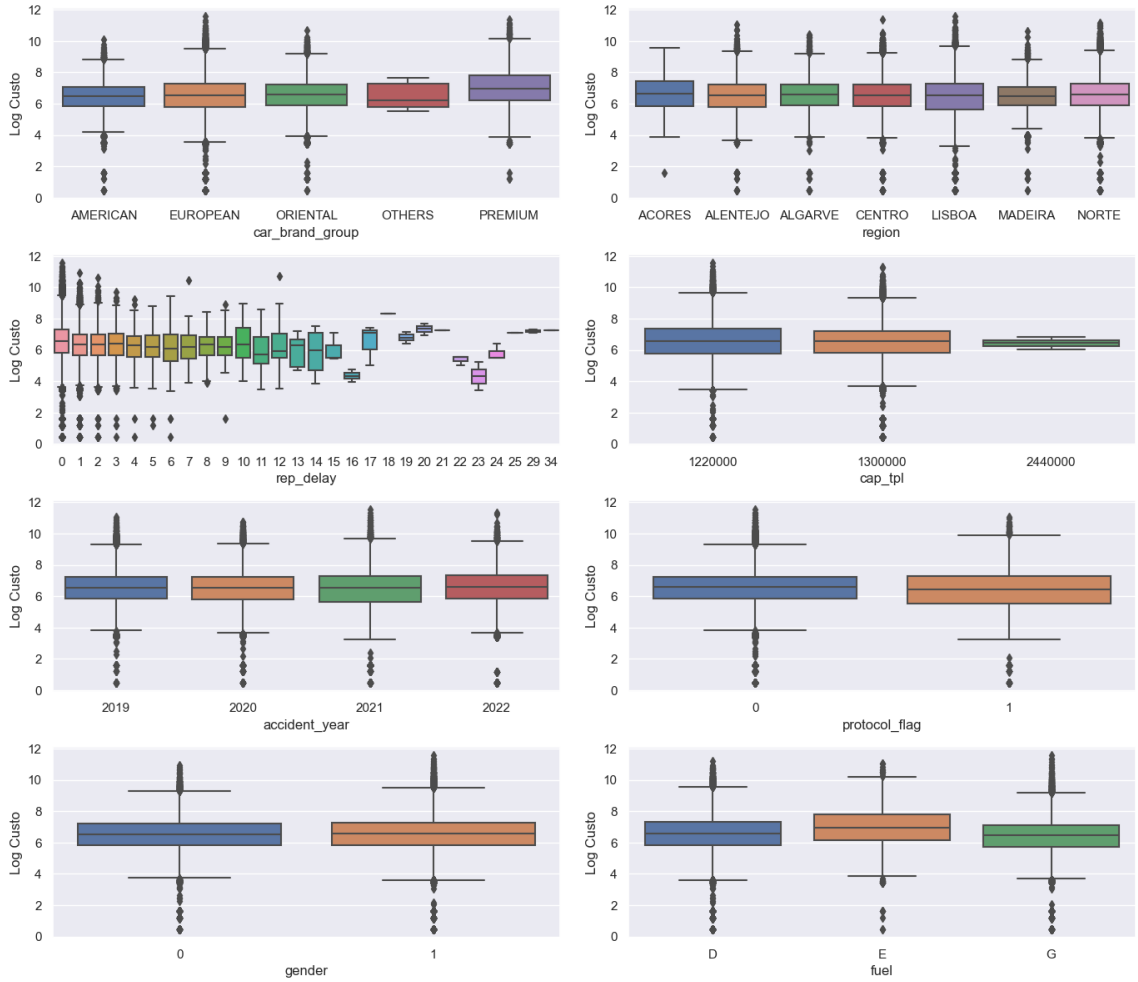


FIGURE 3.4: Cost boxplot of categorical features.

The most granular data about the car insured is the car model, followed by the car brand, and then by the car's brand groups, that was built from the car brand. The most important difference comes from one specific group, the "premium" car brands, which did not consider the models. For this category, the distribution has higher amounts, the median is higher than the other categories. It could suggest that this feature could be transformed into a boolean between premium groups and non-premium groups.

The region was created from the district feature since there are many distinct districts and some of them had very few observations- Therefore it represents the residence zone the where insured person lives. The zone with the least observations is ACORES with 239 observations. All the other zones have more than 2 000 observations. However, the zone of residency does not really seem to differentiate the target feature.

It could be interesting to see how the zone where the accident happened relates to the average cost to check how it influences the target, although this feature is not available.

The accident location or the insured's residence may also influence the choice of a repair shop for the vehicle. While the country's region does not appear to have a significant impact, it is worth noting that whether the insured opts for a partner auto repair shop can affect the average cost. This is because the affiliated network offers special rates for the Insurance Company.

Taking this into consideration, the analysis can be revisited to assess the potential impact of the region. Furthermore, this information may serve as a basis for determining whether a separate regression model for partner repair shops is necessary or not.

3.2 Claims Dataset

The initial datasets provided had suffered modifications, and many of the features available did not give useful information. Feature engineering played an important role there because the features became more insightful. Some features were initially designed to aid in data analysis and exploration; however, for the model, they needed to be handled differently, such as through One Hot Encoding.

To summarize it, Table 3.3 provides the features used to build the model. This table shows a fictitious example of values for each feature.

TABLE 3.3: Features in claims dataset

Features	Description	Example
cost	target feature in log scale - numerical	7,18
eng_disp	engine displacement - numerical	1368
cob0_2	50M€capital - boolean	0
cap_tpl	third-party liability capital - numerical	1300000
accident_year	occurrence year of the claim - numerical	2020
cap_od	own damage capital - numerical	20650
premium	insurance premium - numerical	387,15
codbonus2	bonus - boolean	0
protocol_flag	protocol with professional associations - boolean	0
gender	driver's gender - boolean	0
formacob2	billing method - boolean	1
policy_duration	policy duration - numerical	3
license_time	license duration - numerical	35
driver_age	driver's age - numerical	54
car_age	car's age - numerical	3
rep_delay	reporting delay - numerical	6
regionacores	Açores Region - boolean	0
regionalentejo	Alentejo Region - boolean	0
regionalgarve	Algarve Region - boolean	0
regioncentro	Central Region - boolean	0
regionlisboa	Lisbon Region - boolean	0
regionnorte	North Region - boolean	1
num_lugares2	number of car's seats - boolean	0
car_brand_american	american cars - boolean	1
car_brand_european	european cars - boolean	0
car_brand_oriental	eastern cars - boolean	0
car_brand_others	others cars - boolean	0
car_brand_premium	premium cars - boolean	0
fueld	diesel cars - boolean	0
fuelg	gasoline cars - boolean	1
fuele	electric cars - boolean	0

Chapter 4

Experimental Study

Once the Claims dataset for our case study is established, in this chapter, we will elucidate the experimental study that was undertaken. This study encompasses the feature selection process and the performance evaluation of various machine learning models obtained through parameter tuning.

4.1 Experimental Setup

Considering the nature of our challenge, the most suitable approach is supervised learning, particularly regression. This choice stems from our goal of predicting ultimate claims cost and reserves. While various algorithms could be considered, we've opted for tree-based algorithms due to the non-normal distribution of the target feature and its nonlinear relationship with covariates. For example, for this situation, linear regression seemed a good option but it was discarded considering the nonlinear relationship.

We specifically selected Random Forest, XGBoost, and Gradient Boosting as these ensemble methods provide advantages. They are not constrained by specific data distributions, deliver excellent results without excessive computational demands, and align with techniques to interpret feature importance. Understanding feature importance is crucial, especially in the insurance industry, given its highly regulated nature.

Having 20 features seemed too much, then it came to the necessity to resort to feature selection techniques to understand the true importance of all features for the model and to prevent unnecessary features from remaining and hindering the model's performance.

The hyperparameters of the chosen models are not intuitively selected, therefore it is extremely important to use a hyperparameter tuning technique to determine the hyperparameters that contribute to the best possible performance of the model. After finding the most suitable hyperparameters, a 10-fold cross-validation with 3 repeats is used to train the model.

4.1.1 Feature Selection

Feature selection is a fundamental step in many machine learning pipelines, especially in this case where there are many features and it is very likely that not all of them are relevant to the model, by analyzing the exploratory data analysis it is possible to have an idea but it is still important to use other approaches to be more assertive choosing the best features. It allows to simplify the problem and reduces unnecessary noise.

Before trying the following methods the training data was standardized with Min Max Scaler, but the results were not significantly better than the results without de standardization, therefore this step was eliminated and it was proceeded using data without standardization.

The methods chosen for this step were: Boruta, Select K-Best, Recursive Feature Elimination (RFE), and Lasso Regression. These methods will be briefly presented.

Boruta

This method based on shadow features and binomial distribution, which makes it statistically grounded and not depend on any specific input by the user [28]. In practice, there is a new data frame based on the original one, called a shadow data frame with duplicated features with different values, it is possible to fit a model, Random Forest Regressor in this case, a threshold is defined, and the highest feature importance is kept. After some iterations, it is possible to accept or refuse the features based on probabilities.

Select K-Best

This method selects the features according to the k features with the highest score. It takes as a parameter a score function, like f_regression, for regression problems like this one, and then the algorithm retains the first k features with the highest scores returned by the function.

The f_regression score function starts with a constant model, M_0, then tries all models M_1 consisting of just one feature and pick the best according to the F-value, which is calculated to test if a feature is significant to the regression model. Then try all models M_2 consisting of M_1 plus one other feature and pick the best [29].

This approach was also used with the mutual information function adapted for continuous target variables.The mutual information between two random variables is a non-negative value that measures the dependency between variables. If the value is zero, the variables are independent, and higher values suggest stronger dependency. The function is based on nonparametric methods based on estimation of entropy from k distances between neighbours, as described in [30] and [31]. Both methods are based on the ideas originally proposed in [32]

Recursive Feature Elimination

This method searches for the most optimized subset of features to include in a model. A subset of features is then created in a way that during each iteration the feature with the worst performance is dropped. As a consequence, the subset always gets smaller until the optimal subset is reached [33]. The algorithm chosen was the Random Forest Regressor, with 10-fold-cross-validation and the most important five features selected.

Lasso Regression

This method is a type of linear regression, with L1 regularization in order to reduce overfitting and generalize the model by adding a penalty term. L1 regularization involves adding a term to the objective function that is proportional to the absolute value of the coefficients of the features. Lasso regression proves especially valuable for feature selection due to its L1 regularization term, which encourages shrinking the coefficients of less significant features to zero, effectively removing them from the model. This leads to sparsity in solutions, ensuring that the final model incorporates only the most crucial features. Lasso regression aids in enhancing the performance of a linear regression model by mitigating overfitting and enhancing model generalization [34]. In essence, Lasso regression falls under the umbrella of shrinkage techniques and serves to enhance the predictive performance of linear regression models. It shines in scenarios with more features than observations or when you need to pinpoint a subset of critical features within a larger dataset.

The feature selection approaches used during this project resulted in five sets of features, and it included the set of features with all features to test. In Table 4.1 all the resulting sets of features are displayed. Boruta considered significant only two features, the number of features chosen for Select K Best was 5, which was also considered in RFE since this methodology ranks the features and does not really eliminate features.

TABLE 4.1: Feature selection results

Feature	Boruta	Mutual Information	F-Regression	Lasso Regression	RFE
eng_disp		x	x	x	x
cob_02				x	
cap_tpl				x	
accident_year		x		x	
cap_od	x	x	x	x	x
premium	x	x	x	x	x
codbonus2				x	
protocol_flag				x	
gender				x	
formacob2				x	
policy_duration				x	
license_time				x	
driver_age					x
car_age		x	x	x	
rep_delay				x	x
regionacores				x	
regionalentejo				x	
regionalgarve				x	
regioncentro				x	
regionlisboa				x	
regionnorte				x	
num_lugares2				x	
car_brand_american				x	
car_brand_european				x	
car_brand_oriental					
car_brand_others					
car_brand_premium			x	x	
fueld				x	
fuelg				x	
fuele				x	

In a preliminary way, all six sets of features were used to train three models Random Forest, XGBoost, and Gradient Boosting with no parameter tuning. The three models were initialized with the same seed, and for XGBoost the objective function chosen was squared error. The training was evaluated with 10-fold cross-validation with 3 repeats and with the RMSE.

The RMSE is calculated as follows:

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

Where: N - Total number of observations,

y_i - Actual target value for observation i, and

$\hat{y}_i$ - Predicted target value for observation i.

Those three models were chosen because of the trade-off between interpretability and efficiency since the insurance industry is very regulated. The RMSE results obtained are presented in Table 4.2.

Considering that the results are very similar and oscillate between the models, there is no benefit in keeping all features and the set of features chosen is the one selected by RFE because it only has numerical features, which contributes to the use of more methodologies later and it is coherent in a business point of view. Those are features with characteristics that have influence also from a pricing point of view and may help describe the claims cost.

TABLE 4.2: RMSE training results from model without hyperparameter tuning

Feature Selection Method	RMSE (mean ± standard deviation)		
	Random Forest	XGBoost	Gradient Boosting
All Features	831 ± 7.55	829 ± 20.50	751 ± 3.31
Boruta	1 027 ± 22.11	761 ± 13.16	725 ± 5.01
Mutual Information	899 ± 14.49	782 ± 20.55	728 ± 4.66
F-Regression	923 ± 14.77	778 ± 13.94	728 ± 3.89
Lasso Regression	832 ± 6.68	832 ± 30.33	752 ± 3.36
RFE	841 ± 7.88	789 ± 10.60	739 ± 3.36

4.1.2 Hyperparameter Tuning

Bayesian Optimization provides a principled technique based on Bayes Theorem to direct a search for a global optimization problem that is efficient and effective. It works by

building a probabilistic model of the objective function, called the surrogate function, that is then searched efficiently with an acquisition function before candidate samples are chosen for evaluation on the real objective function [35].

Bayesian Optimization is often used in applied machine learning to tune the hyperparameters of a given well-performing model on a validation dataset. It involves creating a probabilistic model of the objective function and using this model to suggest the next set of hyperparameters to evaluate. This process is repeated until the optimal hyperparameters are found. Bayesian optimization is a more efficient way of searching the hyperparameter space compared to grid search or random search [36].

This optimization was chosen to do the hyperparameter tuning of the three models selected, Random Forest, XGBoost, and Gradient Boosting using the RFE set of features, it is quite fast when compared to other common approaches such as Grid Search, and has been quite used in the later literature [37].

The validation strategy for the optimizer was Repeated K-Fold with 5 splits and 3 repeats, the regressor used was the XGBRegressor. Three searching spaces were defined, one for each model and the respective parameters. Following that, the models were trained using the hyperparameters found using Bayes Optimization [38]. The Table 4.3 displays the searching space used. The step taken between values is automatically determined by the algorithm following a uniform distribution. When there is a decimal point, it looks forward real numbers interval, and when there are not, it ranges between integers.

TABLE 4.3: Hyperparameter search spaces

Hyperparameter	**XGBoost**	**Random Forest**	**Gradient Boosting**
Learning Rate	[0.2, 0.5]	-	[0.01, 0.2]
Max Depth	[2, 4]	[2, 10]	[2, 10]
Subsample	[0.8, 1.0]	-	[0.1, 1.0]
Colsample by Tree	[0.8, 1.0]	-	-
Reg Lambda (L2)	[30, 50]	-	-
Reg Alpha (L1)	[30, 50]	-	-
N Estimators	-	[10, 200]	[10, 200]
Min Samples Split	-	[2, 10]	[2, 10]
Max Features	-	[0.1, 1.0]	[0.1, 1.0]

4.2 Performance Estimation

The first approach, the Baseline scenario is built by splitting the original data set between train, with closed claims and accident years from 2019 to 2021, and test with closed claims that occurred in 2022. Claims with cost bellow 10€ were discarded because it is due to costs with information and should not be considered.

The model was trained using cross-validation with 10 folds and 3 repeats using the hyperparameters found with the Bayesian Optimization approach, and the validation metric chosen to present here is the RMSE (Root Mean Squared Error) since it is easier to interpret.

The RMSE results during training are presented in Table 4.4.

TABLE 4.4: RMSE results - Training set - Baseline

Random Forest	XGBoost	Gradient Boosting
732 $\pm$ 2.61	745 $\pm$ 3.56	779 $\pm$ 13.55

Also, a Wilcoxon-Test was performed between the best result, Random Forest, comparing to XGBoost and Gradient Boosting, to check if the results are statistically different. The test results showed that the results are different with a 95% level of confidence [39].

The feature importance can be analyzed in Figure 4.1. This figure reveals how much each feature contributes to the model's predictions. In Random Forest, feature importance is determined by measuring how effectively each feature reduces prediction error across multiple trees within the ensemble. Features that consistently reduce error are assigned higher importance scores. As shown in the figure, premium stands out as the most influential feature when estimating costs [40].

When it comes to test results, we have the following RMSE results in Table 4.5:

TABLE 4.5: RMSE results - Test set - Baseline

Random Forest	XGBoost	Gradient Boosting
2 142	2 137	**2126**

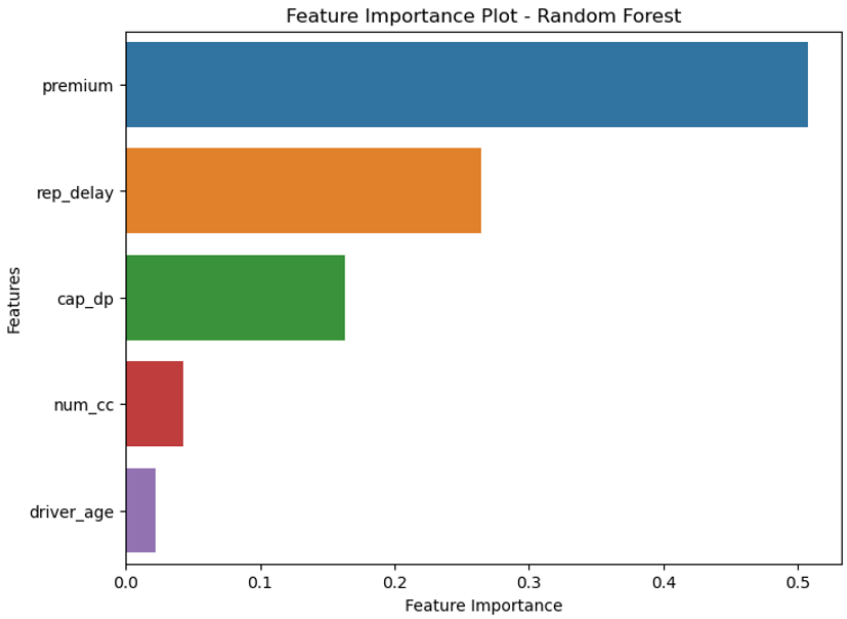


FIGURE 4.1: Baseline model's feature importance - Random Forest.

The test results are significantly worse than the training results, and during test Gradient Boosting performed better than Random Forest. Then we move forward considering Gradient Boosting and we can see the scatterplot in Figure 4.2. It does not show a linear behavior as it should be desirable.

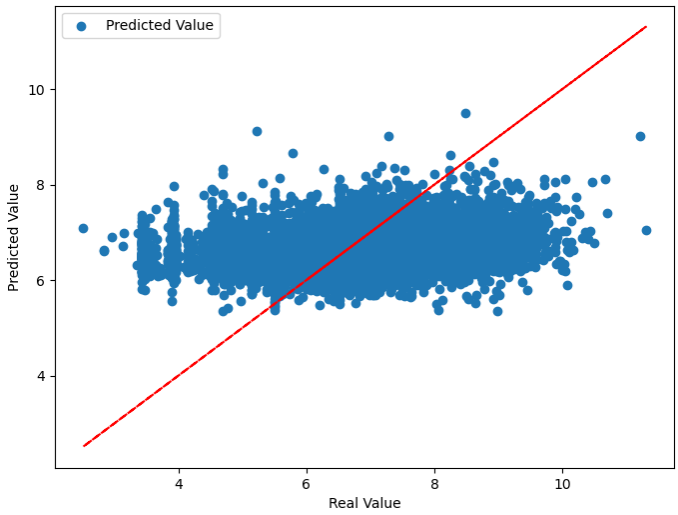


FIGURE 4.2: Real cost vs. expected cost in log scale - Gradient Boosting - Baseline.

Figure 4.3 shows that the shape of the distribution is similar, although the model was not able to extrapolate as much as it should.

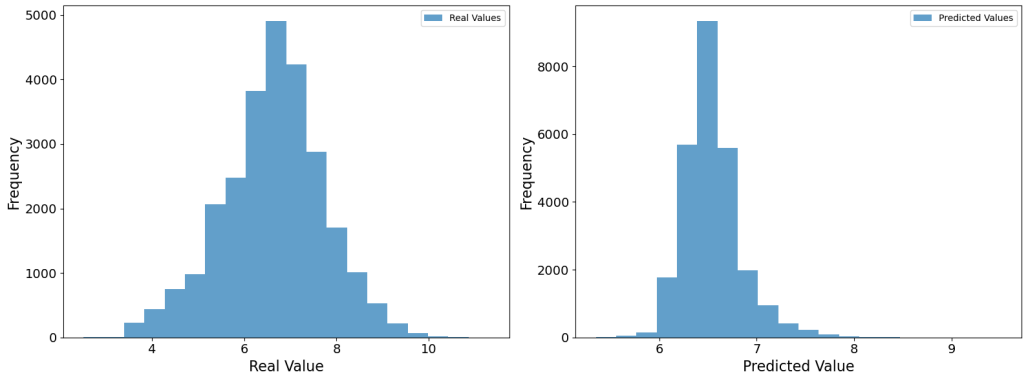


FIGURE 4.3: Histograms for real and expected cost in log scale - Gradient Boosting - Baseline.

By looking at the summary of the results in Table 4.6 we can see that the model gave a total estimative almost 50% below the real sum amount, which is crucial to determine that when comparing this result with Chain Ladder we will have a bad estimative since in real life it would be very bad to have such a underestimated result. Also, the Chain Ladder has a different starting point, therefore it will never estimate such a low value.

TABLE 4.6: Range results - Gradient Boosting - Baseline

	Real Value	Predicted Value	Difference
Minimum	12	210	−8 968
Maximum	81 418	13 495	80 272
Total Sum	36 025 415	19 260 277	16 765 138

The initial idea to compare the regression model with Chain Ladder consisted of doing that by difference, meaning that we would use the closed claims to train the model and for the Chain Ladder as fact, and then estimate the open claims amount. Due to the lack of a directly comparable result, it no longer seems feasible to proceed with the initial idea.

After that baseline model, different scenarios were built to try to improve the model. Those scenarios will be described during this chapter and are summarized in Table 4.7. The baseline scenario is the default and the description highlights the difference from the new scenario to the baseline.

A model with training data capped in 2 500€, which is a business suggestion for a suitable amount, this scenario is referred to as CostsBelow2500. A 10 000€ threshold was also considered but did not improve the model.

TABLE 4.7: Scenarios descriptions

Scenario	Description
Baseline	Baseline version
CostsBelow2500	Train set with costs capped in 2500€
AllFeatures	All features
IDSClaims	Train set with only IDS claims
EuropeanCars	Train set with only European cars
Balanced	Balanced train set
BalancedOriginalCost	Balanced train tet with the target without log transformation

Also, a scenario with all the features mentioned before, scenario AllFeatures was tested to see if it would improve the model, but it did not show better results than with the selected features.

By that point, there is already an alert that the features might not be characterizing the claims very well, as seen during feature selection and now with the models and not obtaining a very different result from the various approaches.

Other scenarios were tested, considering some business specifications, such as the IDS, from Portuguese *Indemnização Direta ao Segurado*, or Direct Settlement with the Insured. It is a system in Portugal that allows policyholders, regardless of fault, to directly contact their own insurance company to settle property damage claims resulting from accidents involving two vehicles, as long as certain conditions are met, including a maximum damage limit of 15 000€ per vehicle. This scenario is called IDSClaims.

Limiting the claims to IDS related claims would limit bit of severity and the range of the claims, it did not improve much the model during training but during test, when you also limit to only IDS claims, considering it is possible to have that information when you open a claim, it improved significantly RMSE results.

After analyzing these models, it is undeniable that the biggest model errors come from large claims, where the model does not have a feature to clearly identify that the claim is more expensive than usual, and it is also where we have fewer observations, which is expected given the right skewness of the cost feature.

Therefore, to attenuate the skewness of the target feature distribution, an undersampling technique was applied to the training set to reduce the number of observations of "small claims". In Balanced the dataset was limited to two chunks, one ranging between 500 and 2 000 euros, and the other between 2 000 and 10 000, each one with 15 000 observations, which is approximately the original number of observations in the second chunk.

The results are worse than the baseline scenario, which suggests that after all the major problem is the lack of features characterizing the claim or the models used are not the most adequate to solve this problem.

Other scenarios were considered but did not go through further evaluation, like separating a model from a specific car brand, using models like Quantile Random Forest Regression, or combining models with Linear Regression and Random Forest, which could be more suitable for unbalanced problems.

From a business perspective, when dealing with the ultimate cost problem, it became apparent that the model tended to predict lower values. In this context, it is more advantageous to lean towards overestimation rather than underestimation. To address this, the final scenario, BalancedOriginalCost was tested without applying the log transformation to the target feature. This adjustment indeed led to higher predictions, aligning better with the business requirements. However, it also resulted in a higher RMSE, indicating a less favorable outcome from a data science standpoint.

The final RMSE results during training for all scenarios are presented in Table 4.8

TABLE 4.8: RMSE results for 10-Fold Cross Validation training

Scenario	RMSE (mean ± standard deviation)		
	Random Forest	XGBoost	Gradient Boosting
Baseline	**732** ± 2.61	745 ± 3.56	779 ± 13.55
CostsBelow2500	**526** ± 0.94	532 ± 1.09	538 ± 1.21
AllFeatures	**739** ± 2.73	774 ± 3.43	808 ± 9.67
IDSClaims	**699** ± 2.10	708 ± 2.36	728 ± 6.29
EuropeanCars	**728** ± 3.21	742 ± 3.85	776 ± 9.55
Balanced	**1 933** ± 11.13	1 945 ± 11.79	1 966 ± 13.03
BalancedOriginalCost	1 790 ± 32.85	**1 786** ± 32.32	1 810 ± 31.35

During test, in addition to RMSE, we also utilize other error metrics such as Huber Loss and Lin Lin Loss, because they could give different insights.

Huber Loss

Commonly used loss functions for regression include $L1(x) = |x|$ and $L2(x) = \frac{1}{2}x^2$. Each of these functions has its own advantages and drawbacks. $L1$ is less sensitive to outliers in the data but lacks differentiability at zero. On the other hand, $L2$ is differentiable everywhere but is highly sensitive to outliers [41]. To address these trade-offs, Huber proposed a compromise loss function denoted as $H_\alpha(x)$ (Equation 4.1):

$$H_\alpha(x) = \begin{cases} \frac{1}{2}x^2, & |x| \leq \alpha \\ \alpha|x| - \frac{1}{2}\alpha^2, & |x| > \alpha \end{cases} \tag{4.1}$$

Here, $\alpha \in \mathbb{R}^+$ is a positive real number that governs the transition from $L1$ to $L2$. The Huber loss offers both differentiability across its range and robustness to outliers. However, one drawback is that the parameter α must be carefully selected [41]. For the purpose of this work, $\alpha = 1$. This loss function is interesting considering the shape of the target distribution and the presence of extreme values that rarely appear.

Lin-Lin Loss Function

This loss function, introduced by Granger [42], is a specialized loss function used in predictive modeling. Unlike traditional loss functions, Lin-Lin exhibits linearity on each side of the origin. However, it distinguishes between positive and negative errors by assigning different penalties due to varying slopes on each side of the origin [43].

Mathematically, the Lin-Lin Loss Function is defined as follows:

$$L(y_{t+h} - \hat{y}_{t+h}) = \begin{cases} a|y_{t+h} - \hat{y}_{t+h}| & \text{if } (y_{t+h} - \hat{y}_{t+h}) > 0 \\ b|y_{t+h} - \hat{y}_{t+h}| & \text{if } (y_{t+h} - \hat{y}_{t+h}) \leq 0 \end{cases} \tag{4.2}$$

Here, $L(\cdot)$ represents a loss function defined for h-step-ahead prediction errors, where y_{t+h} is the actual value of y, $t+h$ periods into the future, and $\hat{y}_{t+h}$ is the predicted value for that same time point. The h represents the lag time for the next prediction, but in this case it is equal to zero. Also a and b are constants to give weight to the functions.

The ratio $\frac{a}{b}$ serves as a measure of the cost associated with underprediction compared to overprediction. For example, if $\frac{a}{b} = 2$, it implies that the penalty for a positive error is twice that of a negative error of the same magnitude. When $a = b$, the average loss aligns with the Mean Absolute Error (MAE) [43].

Table 4.9 shows the results obtained for each of the three metrics chosen in each scenario.

By looking at RMSE results, it indicates the typical error magnitude, ranging from 2 055 to 2 250 across scenarios. Lower RMSE values suggested better overall accuracy, with scenario Balanced having the best performance.

TABLE 4.9: Error metrics during Test

Scenario	RMSE			Huber Loss			Lin Lin Loss		
	Random Forest	XGBoost	Gradient Boosting	Random Forest	XGBoost	Gradient Boosting	Random Forest	XGBoost	Gradient Boosting
Baseline	2 142	2 138	**2 126**	953	**950**	954	**268 148**	279 197	308 962
CostsBelow2500	2 237	**2 235**	**2 235**	998	**995**	**995**	**140 316**	143 727	147 164
AllFeatures	2 136	2 120	**2 107**	946	**941**	942	**272 272**	300 563	337 534
IDSClaims	2 142	2 139	**2 138**	**954**	956	964	**279 408**	292 293	327 285
EuropeanCars	2 143	2 137	**2 128**	954	**952**	959	**267 182**	281 431	346 186
Balanced	**2 055**	2 056	2 068	1 255	**1 252**	1 269	**1 676 168**	1 678 229	1 771 022
BalancedOriginalCost	**2 224**	**2 224**	2 250	1 584	**1 577**	1 584	**2 474 698**	2 472 921	2 531 043

Huber Loss, balancing robustness and differentiability, achieved values between 946 and 1 255 in most scenarios. Baseline showed the lowest Huber Loss, indicating robustness to outliers, while the Balanced exhibited increased sensitivity to extreme values.

Lin-Lin Loss, distinguishing between positive and negative errors, displayed varying performance (140 316 to 2 531 043) across scenarios. Provided the best results in CostsBelow2500 but had higher values in Balanced and BalancedOriginalCost.

The results for BalancedOriginalCost illustrate how RMSE is not sensitive to the error variation, Huber Loss and Lin Lin Loss presented an oscillation much stronger than RMSE when the log scale was removed, increasing the volatility.

In summary, RMSE assessed general accuracy, Huber Loss emphasized robustness and Lin-Lin Loss addressed error types.

4.3 Interpretability Analysis

It is very important to understand the impact of the features in the final results, that is why we are using the SHAP method to help explain the predictions [44].

Even though in almost all of the scenarios Random Forest models had the best results during training, during test, Gradient Boosting had the best performance. Therefore, we are going to apply SHAP to Baseline considering the Gradient Boosting results.

First, it is possible to have a global explanation of the model by analyzing the summary Figure 4.4. This plot provides insights into the most significant features. For each feature, the mean SHAP value is computed across all observations. Notably, the mean of the absolute values is calculated to prevent positive and negative contributions from canceling each other out. The resulting bar plot illustrates this information, with one bar representing each feature. As an example, it is observable that the premium, representing

the premium paid for the insurance, has the highest average impact on the model. And rep_delay, representing the reporting delay, comes just after.

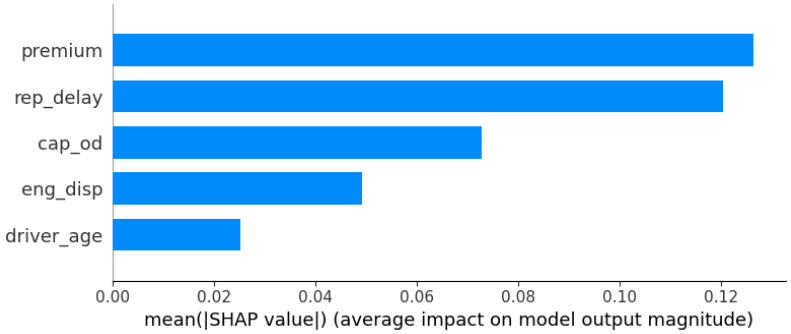


FIGURE 4.4: SHAP's summary barplot - Gradient Boosting - Baseline.

It's evident that both premium and rep_delay will exert significant influence on the prediction outcomes. However, at this stage, we don't have insight into the specific direction in which these features affect the predictions. To gain a better understanding in this regard, we can utilize the beeswarm plot.

The beeswarm Figure 4.5 visualizes all the SHAP values. Along the y-axis, values are grouped by feature, and for each group, the color of the points is determined by the feature value (where higher feature values appear redder). Similar to the mean SHAP plot, the beeswarm plot can be employed to highlight significant relationships, the features are arranged based on their mean SHAP values.

Considering Figure 4.5, and premium as an example, as the feature value increases, the SHAP values also increase. This pattern mirrors what we observed in Figure 4.4, indicating that larger values for the premium result in a higher predicted claims amount.

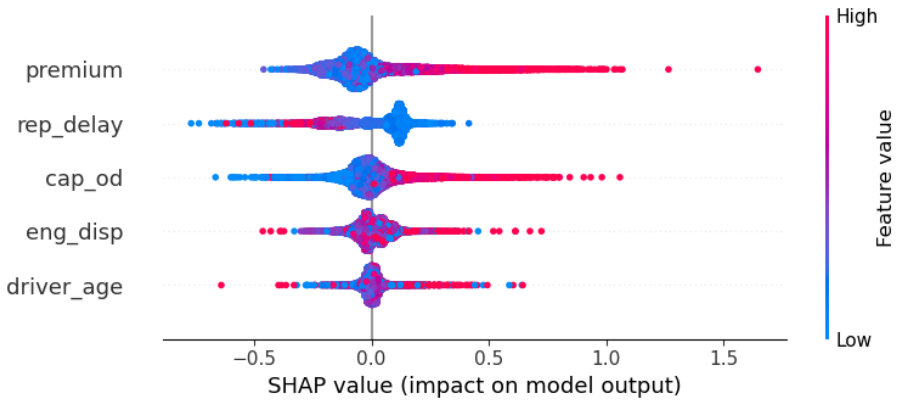


FIGURE 4.5: SHAP's beeswarm plot - Gradient Boosting - Baseline.

Then, we can move to a local explanation, where it is possible to analyze one observation at each time, and how it behaves compared to the mean estimation in Figure 4.6.

We are starting with the observation that represents the biggest error the model has, which is also an unexpectedly high value, and could be considered an outlier but since it is actually possible to have these large events we opt not to remove it.

We have $E[f(x)] = 6.528$ representing the average predicted claim amount in the log scale, while $f(x) = 7.044$ is the predicted claim amount for this particular observation. The SHAP values are all the values in between.

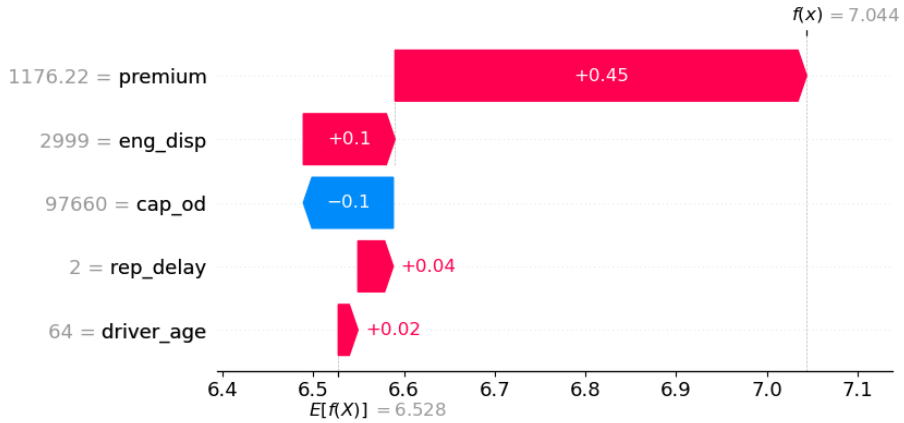


FIGURE 4.6: SHAP's waterfall plot - highest error Gradient Boosting - Baseline.

A notably high value for the premium feature played a crucial role in predicting a cost significantly above the average. In this instance, the model predicted a cost of 1 145,74€, while the actual cost amounted to 81 418,18€. Interestingly, this scenario saw a 0.45 increase in the logarithm of the claim amount.

An alternative method for visualizing these insights is by employing a force plot in Figure 4.7. It can be thought of as a concise version of the waterfall Figure 4.6. We begin from the common base value of 6.528, in logarithmic scale, which is the expected value also presented in Figure 4.6. It allows you to observe how each feature has contributed to the ultimate prediction of 7.044, in log, which is 1 145,96€.

FIGURE 4.7: SHAP's force plot - highest error Gradient Boosting - Baseline.

This Figure 4.7 reinforces the idea of the Figure 4.6 that four of the five features are contributing to increasing the expected amount and only cap_od is contributing to decreasing it.

4.4 Discussion

A central part of this study involved evaluating model performance across various scenarios. Different data preprocessing and feature selection techniques were employed to prepare and exploit the best from the dataset. The models were assessed using different performance metrics: RMSE, Huber Loss, and Lin Lin Loss.

Regarding algorithm performance, Gradient Boosting consistently yielded slightly better results in terms of RMSE, while XGBoost was the best from the Huber Loss point of view, and Random Forest excelled in terms of Lin Lin Loss. This implies that the choice of algorithm may depend on which metric should be more adequate.

In addition to performance evaluation, we considered the explainability of model predictions. Understanding the rationale behind predictions is crucial, particularly in the insurance sector, where model-driven decisions require justification.

The interpretability analysis with SHAP and feature importance helped to understand the model's estimates. It became evident that certain features, like premium, played a significant role in determining claim cost predictions.

However, we also acknowledged that achieving full explainability in complex models, as used in this study, remains challenging. While we can identify important features, comprehending the underlying logic of predictions in detail is a complex task.

All those analysis suggested that it would be extremely important to have stronger features that characterize the degree of damage. The choice of the model played a significant role, maybe combinations of models could help the performance. Also, other techniques to balance the data could be further investigated.

Chapter 5

Conclusions

Here we summarize the main contributions of this project and acknowledge its limitations. Our work offers a different perspective, focusing on specific features related to the claims, which has provided insights for the field. While our project emphasizes the significance of tailored features and the potential of data science in actuarial sciences, it also acknowledges the challenges of comparing traditional and modern methodologies.

5.1 Main Contributions

This project presented a Machine Learning based version of a traditional problem, further analyzed the impacts, and opened space to more exploration. It contributed to the research into the actuarial sciences field with a data science approach and also generated important insights about the available data and how important to have specific features because even though nowadays we have a large amount of data available, sometimes it is not adequate to solve a specific problem.

The results of our experimental study on claims reserving revealed that sometimes too much detail is not helpful. Such is the case for the car's brands groups and the zone of residency, the vehicles could be classified between premium or not, and the zone of residency is not as important as where the car is repaired (partners repairs shops or not).

It is also an important contribution for the company because there are not yet many projects exploring new technologies in the financial areas. The data came from different sources and steps like data cleaning and data engineering open space to further investigate which kind of instability data may present. It opens space for a discussion about collecting different data that could help improve or build different models.

This study has a financial focus, but it can also be exploited by operational areas with the intention of predicting the cost to release workload from the claim manager. It could give a more accurate overview of the operations.

5.2 Limitations and Future Work

The Chain Ladder method is a traditional actuarial approach used to estimate reserves of claims based on historical trends in claim development. It relies on the differences between past and delayed payments to forecast future events. However, the Chain Ladder method does not account for external features or other characteristics that could influence claims.

On the other hand, regression models can take into consideration multiple features and features to predict future values. However, it does not operate in the same manner as the Chain Ladder method regarding the prediction of reserve. This method is based on differences in past patterns and delayed payments.

Given these disparities, conducting a direct comparison between the outcomes generated by the Chain Ladder method and the machine learning methodologies can present challenges and limitations for meaningful comparison in the project.

Machine learning techniques and applications are being released and developed through different industries, and testing and applying them in the finance field is very important to develop an industry that has been known for its traditional methodologies. New tools that help decision-makers to be able to make more data-driven decisions in such an unstable environment are certainly very valuable.

The aim of this work was never to underestimate reliable methodologies that have been used for years but to give others perspective and more information to make the decisions to be more data-driven.

The results presented contribute to the research with not very complex models, that can be explained with the auxiliary methods such as SHAP and replicated.

As we are dealing with a real dataset, it comes with some additional challenges. The dataset provided for this work only has information from 2019, because the data for claims before this date suffered a system migration and might have some inconsistency that would require time to analyze, but more years could be very interesting to check the methodologies in chunks. On the other hand, a very large dataset would cause computational resource constraints.

It was not possible to separate the COVID-19 effect, even though it did not seem to affect the characteristics of the claims, it affected the frequency, and it is not possible to be sure of its impact. It would be interesting to repeat the methodologies after a few years to analyze it further.

Although there are many features available and they are very important, there is a lack of features to really characterize the claim, such as claim type, that could represent, for example, if it is glass breakage, theft, total loss, or other, that kind of feature could help the model to better learn the range of the cost. Also, features about where the insured chose to repair the car, how many days of replacement car were given, or where the accident happened could bring interesting insights and be helpful to train the model.

It would also be interesting if other models were built for other tasks, such as predicting how long the claim takes to be closed, how long after reported it will be paid, and others. Tree-based models were selected due to their simplicity and versatility. However, the limitation of these models in extrapolation became significant. In reality, while large claims are infrequent, they are essential and must be taken into account. Therefore, it would be beneficial to replicate this work with alternative models. Despite some previous projects utilizing neural networks that did not yield favorable results, this remains a sensitive topic deserving further analysis.

As a suggestion for future work, this model could be exploited as an imbalanced regression where the focus is to obtain robust models for extreme values [45].

Appendix A

Features Description

TABLE A.1: Features - Claims Dataset

Feature	Description
Num_Apolice	Policy number related to the claim
ID_Sinistro	Claim number
Cobertura	Type of coverage
Tipo_Gestao	Type of claim
DAS	Date the claim was accepted
Sestado	Status of the claim (open or closed)
cost	Total Claim Cost (target)
OPPAGAS	Amount Paid
Pdtsin	Accident Date
accident_year*	Claim's Accident Year
closing_date*	Claim's Closing Date
closing_year*	Claim's Closing Year

TABLE A.2: Features - Policy Dataset

Feature	Description
Num_Apolice	Policy number related to the claim
codbonus2*	Bonus-Malus
DAT_MATRICULA	Car's Registration Date
DAT_ANO_CONSTRUCAO	Car's Construction Year
NUM_LUGARES	Car's Number of Seats
formapa	Payment Method
formacob	Billing Method
CVE_ZONA_TARIFARIA	Fare Zone
region*	Insured's Zone of Residency
protocol	Protocol (Professional Association)
protocol_flag	Protocol (Professional Association)
CVE_MODELO	Car's Model
eng_disp	Engine Displacement
fuel	Car's Fuel Type
car_brand	Car's Brand
car_brand_group*	Car's Brand Group
cap_tpl	Third-Party Liability Coverage Capital
cap_od	Own Damage Coverage Capital
premium	Insurance Premium Amount
ins_deductible	Insurance Deductible
cob0_2*	Extra Coverage up to 50M
codsitu	Policy Situation
driver_age*	Driver's Age
car_age*	Car's Age
gender*	Driver's Gender
policy_duration*	Policy's Duration

TABLE A.3: Features - Join Data Dataset

Feature	Description
capdp_classes*	Own Damage Capital in Bins
premium_classes*	Insurance Premium in Bins
cc_classes*	Engine Displacement in Bins
age_classes*	Driver's Age in Bins
license_classes*	License's Time in Bins
carage_classes*	Car's Age in Bins
rep_delay*	Reporting Delay

Bibliography

[1] R. L. Bornhuetter and R. E. Ferguson, "The actuary and ibnr," in *Proceedings of the casualty actuarial society*, vol. 59, no. 112, 1972, pp. 181–195. [Cited on page 2.]

[2] T. Mack, "Distribution-free calculation of the standard error of chain ladder reserve estimates," *ASTIN Bulletin: The Journal of the IAA*, vol. 23, no. 2, p. 213225, 1993. [Cited on pages 6 and 11.]

[3] G. Rebala, A. Ravi, and S. Churiwala, *Machine Learning Definition and Basics*. Cham: Springer International Publishing, 2019, pp. 1–17. [Online]. Available: https://doi.org/10.1007/978-3-030-15729-6_1 [Cited on page 7.]

[4] P. Flach, *The ingredients of machine learning*. Cambridge University Press, 2012, p. 1348. [Cited on pages 7 and 8.]

[5] J. Han, M. Kamber, and J. Pei, "Data mining concepts and techniques third edition," *University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University*, 2012. [Cited on pages xi and 8.]

[6] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001. [Cited on pages 8 and 9.]

[7] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794. [Cited on page 9.]

[8] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001. [Cited on page 10.]

[9] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017. [Cited on page 10.]

[10] C. Molnar, "A guide for making black box models explainable," *URL: https://christophm. github. io/interpretable-ml-book*, vol. 2, no. 3, 2018. [Cited on pages 10 and 11.]

[11] J. Crevecoeur, J. Robben, and K. Antonio, "A hierarchical reserving model for reported non-life insurance claims," *Insurance: Mathematics and Economics*, vol. 104, pp. 158–184, 2022. [Cited on page 11.]

[12] M. Aleandri, "Case reserving in non-life practice using individual data and machine learning," 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:198231923 [Cited on pages 12 and 13.]

[13] S. Jamal, S. Canto, R. Fernwood, C. Giancaterino, M. Hiabu, L. Invernizzi, T. Korzhynska, Z. Martin, and H. Shen, "Machine learning and traditional methods synergy in non-life reserving (mltms)," *ASTIN WORKING PARTY*, 2018. [Cited on page 12.]

[14] F. Duval and M. Pigeon, "Individual loss reserving using a gradient boosting-based approach," *Risks*, vol. 7, no. 3, 2019. [Online]. Available: https://www.mdpi.com/2227-9091/7/3/79 [Cited on page 12.]

[15] D. Qiu, "Individual claims reserving: Using machine learning methods," Ph.D. dissertation, 12 2019. [Cited on page 13.]

[16] B. H. champion, R. Gächter, S. Jamal, A. Schaeper, A. Boumezoued, A. McGuinness, E. N. Gustafsson, F. Cuypers, G. Taylor, I. Valdés, K. Krøier, M. I. Silva, M. V. Wüthrich, M. Cairns, N. Rietdorf, P. D. England, R. R. Cerchiara, S. Betz, S. Mueck, T. Korzhynska, T. van den Vorst, and V. Magatti, "Individual claim development with machine learning," 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:198234564 [Cited on page 13.]

[17] A. Gabrielli, "Claims reserving and neural networks," Doctoral Thesis, ETH Zurich, Zurich, 2020. [Cited on page 13.]

[18] M. T. Al-Mudafer, "Probabilistic forecasting with neural networks applied to loss reserving," Ph.D. dissertation, University of New South Wales, 2020. [Cited on page 13.]

[19] P. D. England and R. J. Verrall, "Stochastic claims reserving in general insurance," *British Actuarial Journal*, vol. 8, no. 3, pp. 443–518, 2002. [Cited on page 13.]

[20] M. V. Wüthrich and M. Merz, "Yes, we cann!" *ASTIN Bulletin: The Journal of the IAA*, vol. 49, no. 1, pp. 1–3, 2019. [Cited on page 13.]

[21] A. Gabrielli, R. Richman, and M. V. Wüthrich, "Neural network embedding of the over-dispersed poisson reserving model," *Scandinavian Actuarial Journal*, vol. 2020, no. 1, pp. 1–29, 2020. [Online]. Available: https://doi.org/10.1080/03461238.2019.1633394 [Cited on page 14.]

[22] J. H. L. Poon, "Penalising unexplainability in neural networks for predicting payments per claim incurred," *Risks*, vol. 7, no. 3, 2019. [Online]. Available: https://www.mdpi.com/2227-9091/7/3/95 [Cited on page 14.]

[23] A. Gabrielli, "A neural network boosted double overdispersed poisson claims reserving model," *ASTIN Bulletin: The Journal of the IAA*, vol. 50, no. 1, p. 2560, 2020. [Cited on page 14.]

[24] C. F. d. J. d. Sousa, "Ibnr techniques in health insurance: a machine learning approach," Master's thesis, NOVA Information Management School, 2023. [Cited on page 14.]

[25] M. Waskom, O. Botvinnik, D. O'Kane, P. Hobson, S. Lukauskas, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole, J. Warmenhoven, J. de Ruiter, C. Pye, S. Hoyer, J. Vanderplas, S. Villalba, G. Kunter, E. Quintero, P. Bachant, M. Martin, K. Meyer, A. Miles, Y. Ram, T. Yarkoni, M. L. Williams, C. Evans, C. Fitzgerald, Brian, C. Fonnesbeck, A. Lee, and A. Qalieh, "mwaskom/seaborn: v0.8.1 (september 2017)," Sep. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.883859 [Cited on page 18.]

[26] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. [Online]. Available: https://ggplot2.tidyverse.org [Cited on page 20.]

[27] C. Wissler, "The spearman correlation formula," *Science*, vol. 22, no. 558, pp. 309–311, 1905. [Cited on page 21.]

[28] S. Mazzanti. (2020) Boruta explained exactly how you wished someone explained to you. Accessed on 20/02/2023. [Online]. Available: https://towardsdatascience.com/boruta-explained-the-way-i-wish-someone-explained-it-to-me-4489d70e154a [Cited on page 28.]

[29] Hands-on machine learning with scikit-learn. Accessed on 15/01/2023. [Online]. Available: https://www.educative.io/courses/hands-on-machine-learning-with-scikit-learn/feature-selection [Cited on page 29.]

[30] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E*, vol. 69, p. 066138, Jun 2004. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.69.066138 [Cited on page 29.]

[31] B. C. Ross, "Mutual information between discrete and continuous data sets," *PLoS One*, vol. 9, no. 2, pp. 1–5, Feb. 2014. [Cited on page 29.]

[32] L. F. Kozachenko and N. N. Leonenko, "Sample estimate of the entropy of a random vector," *Problemy Peredachi Informatsii*, vol. 23, no. 2, pp. 9–16, 1987. [Cited on page 29.]

[33] J. Brownlee. (2020) Recursive feature elimination (rfe) for feature selection in python. Accessed on 02/02/2023. [Online]. Available: https://machinelearningmastery.com/rfe-feature-selection-in-python/ [Cited on page 29.]

[34] (2022) Joint feature selection with multi-task lasso in scikit learn. Accessed on 22/02/2023. [Online]. Available: https://www.geeksforgeeks.org/joint-feature-selection-with-multi-task-lasso-in-scikit-learn/ [Cited on page 29.]

[35] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf [Cited on page 32.]

[36] S. Hosseini. (2021) A practical guide to hyperparameter tuning of xgboost models using bayesian optimization and grid search. Accessed on 18/01/2023. [Online]. Available: https://medium.datadriveninvestor.com/introduction-31c985114aa1 [Cited on page 32.]

[37] H. Cho, Y. Kim, E. Lee, D. Choi, Y. Lee, and W. Rhee, "Basic enhancement strategies when using bayesian optimization for hyperparameter tuning of deep neural networks," *IEEE access*, vol. 8, pp. 52 588–52 608, 2020. [Cited on page 32.]

[38] T. Head, M. Kumar, H. Nahrstaedt, G. Louppe, and I. Shcherbatyi, "scikit-optimize/scikit-optimize," Oct. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5565057 [Cited on page 32.]

[39] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020. [Cited on page 33.]

[40] O. Grisel, A. Mueller, Lars, A. Gramfort, G. Louppe, P. Prettenhofer, T. J. Fan, M. Blondel, V. Niculae, J. Nothman, A. Joly, G. Lemaitre, J. Vanderplas, L. Estève, manoj kumar, H. Qin, J. du Boisberranger, N. Hug, N. Varoquaux, R. Layton, J. H. Metzen, and A. Jalali, "scikit-learn/scikit-learn: scikit-learn 1.1.2," Jan. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.7566169 [Cited on page 33.]

[41] G. P. Meyer, "An alternative probabilistic interpretation of the huber loss," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 5261–5269. [Cited on pages 37 and 38.]

[42] C. W. Granger, "Prediction with a generalized cost of error function," *Journal of the Operational Research Society*, vol. 20, pp. 199–207, 1969. [Cited on page 38.]

[43] Y. Ulu, "Optimal prediction under linlin loss: Empirical evidence," *International Journal of Forecasting*, vol. 23, no. 4, pp. 707–715, 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207007000441 [Cited on page 38.]

[44] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and

R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf [Cited on page 39.]

[45] R. P. Ribeiro and N. Moniz, "Imbalanced regression and extreme value prediction," *Machine Learning*, vol. 109, pp. 1803–1835, 2020. [Cited on page 45.]