NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF MATHEMATICS

GONÇALO PEDRO CARAPUCINHA BRANCO DUARTE

Master in Actuarial Mathematics

AUTOMOBILE INSURANCE PRICING WITH MACHINE LEARNING CONTRIBUTIONS

MASTER IN ACTUARIAL MATHEMATICS

NOVA University Lisbon

september, 2024

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTMENT OF MATHEMATICS

AUTOMOBILE INSURANCE PRICING WITH MACHINE LEARNING CONTRIBUTIONS

GONÇALO PEDRO CARAPUCINHA BRANCO DUARTE

Master in Actuarial Mathematics

Adviser: Doutora Gracinda Rita Guerreiro
Associate Professor, NOVA University Lisbon

Examination Committee

Chair: Doutor Luís Pedro Carneiro Ramos
Associate Professor, NOVA University Lisbon

Rapporteur: Doutor Pedro Alexandre da Rosa Corte Real
Associate Professor, NOVA University Lisbon

MASTER IN ACTUARIAL MATHEMATICS

NOVA University Lisbon
september, 2024

Automobile Insurance Pricing with Machine Learning Contributions

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.22) [1].

Acknowledgements

To my professor and mentor Gracinda Guerreiro for her support throughout the months of this project and for her availability.

To my parents for always being there to help me with all my problems and concerns. For giving me the support I needed throughout my academic journey. Thank you for always being there.

To my brothers, Francisco and Inês, for all your support and guidance over the years.

To Rafaela Moura for her support and encouragement throughout this project. For all the energy you have given me to complete this project.

To all my colleagues at UNA Seguros for all the moments of discussion and reflection on the questions raised and doubts throughout this project. To Ana Lourenço, Joana Mascarenhas, Ana Maximino, Pedro Coelho, Paula Marques and Francisca Bacalhau, thank you for everything!

To Eduardo Dias, for trusting me from the beginning and giving me the opportunity to learn every day. Thank you for granting me all the time necessary to complete this project and for always being open to answer my many questions.

Abstract

The challenge addressed in this dissertation arises from the predominant use of the Generalised Linear Model (GLM) to create tariffs structures in the insurance market. With the advance of Machine Learning (ML) techniques, opportunities arise to improve the accuracy and predictive capacity of the models used in the insurance sector. Machine Learning models, such as the Gradient Boosting Machine (GBM), stand out for their robustness in capturing non-linear patterns and complex interactions between variables. However, despite the predictive superiority that these models tend to offer, the challenge of interpretability becomes a significant obstacle. This work aims to contribute to the debate on the incorporation of Machine Learning models in the insurance market, using as a case study the comparison between GLM and GBM in the modelling of claims frequency and severity. The aim is to compare the two models in terms of predictive effectiveness, assessing how each model responds to the dataset, and what ability is to accurately predict future results. It is important to note that in this dissertation we do not aim to create a tariff structure, but rather to analyse the relative performance of the two models. The intention is to provide a solid basis for comparison, enabling a better understanding of how Machine Learning models can complement or improve the GLM outputs. Through this analysis, we hope to contribute to a more informed discussion on the adoption of Machine Learning methods in the insurance market.

Keywords: GLM, GBM, Cluster, Non-Life Insurance, Automobile Insurance, Pricing Structure

Resumo

O problema abordado nesta tese surge do uso predominante dos Modelos Lineares Generalizados (MLG) para a criação de tarifas no mercado segurador. Com o avanço das técnicas de Machine Learning (ML), surgem oportunidades para aprimorar a precisão e a capacidade preditiva dos modelos utilizados no setor segurador. Modelos de Machine Learning, como o Gradient Boosting Machine (GBM), destacam-se pela sua robustez na captura de padrões não-lineares e interações complexas entre variáveis, algo que o MLG, por ser linear na sua estrutura, não consegue captar com tanta eficácia. No entanto, apesar da superioridade preditiva que esses modelos tendem a oferecer, o desafio da interpretabilidade torna-se um entrave significativo. Este trabalho visa contribuir para o debate sobre a incorporação de modelos de Machine Learning no mercado segurador, utilizando como caso de estudo a comparação entre MLGs e GBMs na modelagem de frequência e severidade de sinistros. O objetivo desta dissertação é comparar os dois modelos em termos de eficácia preditiva, avaliando como cada um responde ao conjunto de dados, e qual a sua capacidade de prever resultados futuros com precisão. É importante salientar que esta dissertação não tem como objetivo criar uma tarifa, mas sim analisar o desempenho relativo dos dois modelos. A intenção é proporcionar uma base sólida de comparação, que permita entender melhor como os modelos de Machine Learning podem complementar ou melhorar o modelo GLM. Através desta análise, espera-se contribuir para uma discussão mais informada sobre a adoção de métodos de Machine Learning no mercado segurador.

Contents

List of Figures

List of Tables

Acronyms

ASW Average Silhouette Width

GBM Gradient Boosting Machine
GLM Generalised Linear Model

IBNER Incurred But Not Enough Reported
iid independent and identically distributed

PDP Partial Dependence Plot

RSS Residual Sum of Squares

SHAP SHapley Additive exPlanations

TPL Third Party Liability

WCSS Within-Cluster Sum of Squares

1

Introduction

The aim of this dissertation is to discuss and introduce Machine Learning models into Non-life insurance pricing models. Being the Generalised Linear Model (GLM) the most widely model used in the insurance industry when it comes to setting premiums in the Non-life insurance, Machine Learning models, especially the Gradient Boosting Machine (GBM) model, have been increasingly studied by actuaries. Over the last years there has been a growing number of papers, on the implementation of Machine Learning in pricing in the insurance sector, which has generated results that show that these models generate faster and more predictive outputs when compared to GLM. Nevertheless, the main drawback of using GBM to develop tariff structures is the interpretability of the final model, although this lack of interpretability has been improved through the use of Partial Dependence Plot (PDP), which help to predict and better understand the behaviour of the features.

This dissertation is divided into two main sections, a theoretical part, based on a literature review, followed by a case study. Firstly, the Literature Review chapter describes the application of the various methods used in the practical part, as well as providing a context for the collective risk model. Then, chapter 3 presents the case study of how the project was carried out, explaining all the steps, from data manipulation to comparison of the models used.

2

Literature Review

2.1 Collective Risk Model

The main model used in non-life insurance is the collective risk model, which shifts the focus from individual risks to aggregate losses by ignoring the specific details on how many risks there are, the individual probability and severity of loss for each risk [2]. Instead, it assumes that there will be a number N of claims in each year, where N is a discrete random variable with distribution F_N:

$$N \sim F_N(\lambda_1, \lambda_2, ...)$$

where $\lambda_1.\lambda_2...$ are the parameters of the claim count distribution. Some commonly used frequency distributions in this context are:

1. Poisson, which has a single parameter, the rate, where it represents the expected number of claims.

2. Negative Binomial, which has two parameters, the rate and the variance-to-mean ratio.

By focusing on aggregate losses rather than individual risks, the collective risk model simplifies analysis and provides a practical approach to non-life insurance pricing and reserving.

The severity of each loss, denoted as X_j, which is independent and identically distributed (iid) for all losses, is assumed to follow a common probability distribution, F_X. The severity distribution is characterised by one or more parameters and can take various forms such as Log-normal, Gamma or Pareto.

The collective risk model assumes that the number of losses, N, and the severity of losses, X, are independent. This means that the severity of a given loss does not depend on the number of losses in a given period. By modelling the frequency and severity of losses separately and assuming their independence, the collective risk model provides a framework for analysing and quantifying total losses in non-life insurance[2].

According to the collective risk model, the total losses can be expressed as:

$$S = X_1 + ... + X_N \tag{2.1}$$

where N is the number of losses in a given year, and X_j is the amount of a given loss, where $j = 1, ..., N$.

The expected total amount of losses in a given year can be calculated by the following equation:

$$\mathbb{E}(S) = \mathbb{E}(N)\mathbb{E}(X). \tag{2.2}$$

The variability of the total losses, measured by the variance, is given by:

$$\mathbb{V}(S) = \mathbb{V}(X)\mathbb{E}(N) + \mathbb{E}(X)^2\mathbb{V}(N). \tag{2.3}$$

Formula (2.3) shows that the variance of the total losses depends on both the variance of the individual loss amounts and the variance of the number of losses.

In summary, the collective risk model helps to predict the total amount of losses (aggregate losses) that an insurer can expect. It also shows how much these losses might fluctuate and allows a deeper analysis of the overall risk. This information is essential for effective risk management and analysis.

2.2 Frequency Modelling

The primary objective of frequency modelling is to develop a statistical model that can accurately predict the expected number of losses occurring within a given time period. To build this model, the typical approach is to use historical data on the number of losses and the corresponding exposure (e.g. number of policies or the given time that the policies are insured) for each past period. This data is then used to fit a probability distribution that can capture the underlying pattern of loss events. The Binomial, Poisson and Negative Binomial distributions are commonly used for claim number prediction in insurance companies. They are known as counting distributions, which makes them ideal for predicting the number of claims expected in a given period.

These three distributions have some useful properties: they all belong to the Panjer class, which is characterised by the fact that there exist constants a and b such that:

$$\frac{p_k}{p_{k-1}} = a + \frac{b}{k}, k = 1, 2, 3, ... \tag{2.4}$$

where p_k is the probability of having exactly k losses, and p_0 is uniquely determined by imposing the condition

$$\sum_{k=0}^{\infty} p_k = 1.$$

2.2.1 Poisson Distribution

The Poisson distribution is the most important of the counting distributions in insurance pricing applications.

Given N, a discrete random variable with parameter $\lambda \in \mathbb{R}^+$, commonly called Poisson Rate. The probability of $n \in \mathbb{N}_0$ events occur in a given interval, corresponds to the probability mass funtion:

$$P(N = n) = \frac{e^{-\lambda}\lambda^n}{n!} \tag{2.5}$$

The mean and variance of the number of claims are given by:

$$\mathbb{E}(N) = \mathbb{V}(N) = \lambda.$$

The key assumption is that the rate at which losses occur is constant for every insured within a given time period, due to the properties of Homogeneous Poisson Process. If the rate is not constant, other interesting distributions are produced, such as the Negative Binomial.

2.2.2 Negative Binomial Distribution

The Negative Binomial distribution goes beyond the simplicity of the Poisson distribution. Let N be the random variable, with parameters r, defined as the number of failures up to r successes, and p, the probability of (obtaining a) success. Let x be the number of failures up to r successes. Since Bernoulli's trials are performed with replacement and are independent, the probability of obtaining a sequence in which there are r successes and x failures is:

$$(1-p)^x (p)^r$$

Once $x + r$ Bernoulli experiments have been performed, leading to r successes and x failures, the last of which will lead to a success, i.e. the result of the $(x + r)$-th Bernoulli experiment is always fixed as a success, we thus have

$$\begin{pmatrix} x+r-1 \\ x \end{pmatrix}$$

different possible and equally probable sequences that can lead to the desired result of x failures and $r - 1$ successes in the first $x + r - 1$ Bernoulli trials. With that, we have:

$$P(N = x) = \begin{pmatrix} x+r-1 \\ x \end{pmatrix} (1-p)^x (p)^r, 0 \leq p \leq 1; r \geq 1; x = 0, 1, \ldots \tag{2.6}$$

The mean and variance of the Negative Binomial are given by

$$\mathbb{E}(N) = \frac{r(1-p)}{p}$$

$$\mathbb{V}(N) = \frac{r(1-p)}{p^2}.$$

Nevertheless, the Negative Binomial distribution can be seen as Poisson Process whose Poisson rate follows a Gamma distribution. More specifically, if $N \sim Poisson\,(\Lambda)$, $\Lambda \sim Gamma\,(\theta, \alpha)$, then

$$N \sim NB(p = \theta, r = \alpha).$$

2.3 Severity Modelling

Having met the challenge of predicting loss frequency, we now move on to the next stage of the pricing journey: modelling loss severity. This involves analysing the historical loss experience of the insurance portfolio to understand the distribution of loss amounts.

In practice, claims do not always reach final settlement immediately. There is often a discrepancy between the initial estimate (which includes both paid and outstanding claims amount) and the final settled amount. This difference is known as Incurred But Not Enough Reported (IBNER). When dealing with limited information on IBNER, there are two common approaches: Ignoring IBNER may seem straightforward, but it introduces bias into the analysis. This approach risks inaccuracies because the model may not account for the potential gap between initial estimates and final settlements. On the other hand, by using only closed claims, this approach focuses only on claims that are fully or almost fully settled. While this avoids the uncertainty of the estimates, it may introduce a further bias. Larger claims, which typically take longer to settle, may be under-represented, leading to an underestimation of the true size of potential losses.

The choice of approach depends on the data available and the specific objectives of the analysis. Both methods have their advantages and disadvantages, and understanding these trade-offs is critical to building an accurate severity model.

In our case study, we have a portfolio with only closed claims, so with that we will not apply any extra procedure to our study, but it really depends on data availability what type of decisions to make.

2.3.1 Gamma Distribution

The Gamma distribution is often used to model the distribution of claim sizes because of its flexibility in representing different shapes and behaviors of data. Considering Y a continuous variable which follows a Gamma distribution with parameters $\alpha > 0$ and $\lambda > 0$, the probability density function is defined as the following:

$$f(x) = \frac{\lambda^\alpha}{\Gamma(\alpha)} e^{-\lambda x} x^{\alpha-1} \tag{2.7}$$

where $x > 0$. The mean and variance of Gamma distributions are given by:

$$\mathbb{E}(Y) = \frac{\alpha}{\lambda}$$
$$\mathbb{V}(N) = \frac{\alpha}{\lambda^2}$$

2.4 Clustering

Data classification is a fundamental process in many fields, allowing data points to be grouped based on their distinctive characteristics. Two widely used techniques for this purpose are clustering and discriminant analysis. Clustering algorithms identify inherent groups within data without prior knowledge of class labels, while discriminant analysis uses predefined class labels to build a model for classifying new data points. This dissertation explores the application of the k-means clustering algorithm, a widely used clustering technique that divides data into k clusters.

Clustering is an unsupervised learning method that aims to group similar objects or data points into subsets, called clusters, based on their inherent patterns or structures, without any prior knowledge of labels or classes. The goal is to identify natural groupings in the data and assign objects to clusters based on similarity, usually measured by given distance metrics.

According to (Jain et all 1999) [3], typically the process of pattern clustering entails the following sequential steps: pattern representation and proximity measure, clustering, data abstraction and assessment of output.

The pattern representation phase begins with a careful consideration of the attributes that characterise the objects to be clustered. This step not only involves determining the number and types of attributes, but also involves feature selection and extraction. Conversely, feature extraction involves transforming the original attributes into a new set, potentially increasing the effectiveness of the clustering process by revealing important patterns.

Pattern proximity is usually measured by a distance function. This entails selecting an appropriate distance measure tailored to the data domain. Whether employing Euclidean distance, Manhattan distance, or other metrics, the chosen measure is critical for subsequent clustering operations, guiding the identification of cohesive clusters.

The grouping step can be performed in a number of ways. Output clustering can be either hard, where the data is divided into distinct groups, or fuzzy, where each pattern has a varying degree of membership in different output clusters. There are two types of clustering algorithms: hierarchical and partitional. Hierarchical clustering algorithms create a nested set of partitions by merging or splitting clusters according to similarity criteria. In contrast, partitional clustering algorithms determine the partition that best optimises a clustering criteria.

Data abstraction is a central process involving the extraction of a simplified and concise representation from a dataset. This simplification can serve two purposes: first, to facilitate

efficient post-processing by machines, and second, to enhance human comprehension and intuitive appeal.

A key aspect of the clustering process is to assess the validity and coherence of the clusters obtained. Through rigorous analysis, the clustering results are evaluated against predefined criteria to measure the fidelity of the clustering output.

2.4.1 Dissimilarity Measures

Dissimilarity measures, commonly known as distance measures, are integral to data clustering as almost all clustering algorithms use them to establish clustering criteria. A distance measure, denoted D, is a binary function that satisfies the following conditions:

1. Non-negativity: $D(x, y) > 0$
2. Symmetry: $D(x, y) = D(y, x)$
3. Reflexivity: $D(x, y) = 0$ if and only if $x = y$
4. Triangle inequality: $D(x, z) \leq D(x, y) + D(y, z)$,

where x, y and z are arbitrary data points.

For continuous data, the most used dissimilarity measures are based on Minkowski distance [4], which is defined by:

$$D(x, y) = \left(\sum_{j=1}^{d} |x_j - y_j|^p \right)^{\frac{1}{p}} \quad \text{for } p \geq 0. \tag{2.8}$$

Euclidean distance computes the root of square difference between coordinates of pair of objects. The Euclidean distance is a special case of the Minkowski distance [5], when $p = 2$ and it is defined as:

$$D(x, y) = \left(\sum_{j=1}^{d} |x_j - y_j|^2 \right)^{\frac{1}{2}}. \tag{2.9}$$

On the other hand, the Manhattan Distance computes the absolute differences between coordinates of pairs of objects. The Minkowski distance encompasses the Manhattan distance also as a specific instance[6], with $p = 1$, as it is defined in the following equation.

$$D(x, y) = \sum_{j=1}^{d} |x_j - y_j|. \tag{2.10}$$

The main difference between Euclidean and Manhattan distances is how they measure the dissimilarity between two data points. Euclidean distance is the straight line distance between two points, calculated as the square root of the sum of the squared differences between the corresponding coordinates. In contrast, the Manhattan distance calculates dissimilarity as the sum of the absolute differences between the coordinates. It considers

the absolute differences in each dimension independently, without regard to the overall direction or length of the line connecting the points. The Manhattan distance is more robust to outliers and is suitable when individual coordinate differences are important, regardless of their overall magnitude.

2.4.2 Clustering Taxonomy

Over the decades, numerous clustering algorithms have been proposed, which can be broadly divided into two categories: partitional and hierarchical clustering algorithms. Partitional clustering divides a dataset into a single partition, while hierarchical clustering creates a sequence of nested partitions [7].

When it comes to partitional clustering algorithms, they can be split into two main types: hard clustering and soft clustering. Hard clustering, also called crisp clustering, puts each data point into just one cluster. On the other hand, soft clustering, known as fuzzy clustering, lets data points be part of multiple clusters to different extents.

When it comes to hierarchical clustering algorithms, they can be divided into two main types: agglomerative and divisive. Agglomerative hierarchical clustering takes a step-by-step, bottom-up approach. It starts by treating each data point as its own cluster, and then gradually merges the closest pairs of clusters based on a defined rule until all data points are in one cluster. Conversely, divisive hierarchical clustering uses a top-down strategy. It starts with all data points in a single cluster and then recursively divides larger clusters into smaller ones until each cluster contains only one data point.

This taxonomy provides a systematic framework for understanding the diverse range of clustering algorithms and their fundamental characteristics, serving as a valuable reference for researchers and practitioners in the fields of data analysis and machine learning.[8]

2.4.3 K-Means Algorithm

The k-means algorithm stands out as one of the most commonly used clustering techniques over the past few decades. Its popularity stems from its simplicity and effectiveness. Independently developed by Sebestyen (1962) [9] and Macqueen (1967) [10], k-means aims to minimize variation within clusters, building upon the strategies outlined by Thorndike (1953) [11] , Cox (1957) [12] , and Fisher (1958) [13].

The k-means algorithm divides a given data set into clusters by assigning each data point to the nearest centroid and aims to minimize the total variance within the cluster. It ensures this by iteratively optimizing the positions of the centroids until convergence is reached. This optimization process minimizes the squared Euclidean distance between each data point and its assigned centroid. The algorithm's objective is to find the optimal clustering solution that minimizes the total squared error across all clusters. Given a set of n data points $X = (x_1, x_2, ..., x_n)$ the objective function is described by the following:

$$P(U, Z) = \sum_{l=1}^{K} \sum_{i=1}^{n} u_{il} ||x_i - z_l||^2. \tag{2.11}$$

Let's denote k as the number of clusters chosen by the user. $U = (U_{il})_{nxk}$ is a n x k partition matrix, $Z = \{z_1, z_2, ..., z_k\}$ is a set of clusters centers, also $||.||$ is the L^2 Euclidean distance. The partition matrix U hold the information about the clusters and satisfies the following conditions:

$$u_{il} \in \{0, 1\}, i = 1, 2, ..., n, l = 1, 2, ..., k \tag{2.12}$$

$$\sum_{l=1}^{K} u_{il} = 1, i = 1, 2, ..., n. \tag{2.13}$$

As mentioned above, the k-means algorithm is an approximate algorithm that aims to minimise the objective function. The only inputs to the algorithm are the data set and the number of clusters k. First, the algorithm initialises $z_1, z_2, ..., z_k$ by randomly selecting k points from **X** and then iterates in 3 steps:

1. The calculation of distance between x_i and z_j, $\forall_{1 \leq i \leq n, 1 \leq j \leq k}$;
2. Updating the partition matrix U. Fixing the clusters center $Z = \{z_1, z_2, ..., z_k\}$, then the objective function is minimized

$$\begin{cases} 1, & \text{if } ||x_i - z_l|| = \min_{1 \leq j \leq k} ||x_i - z_l|| \\ 0, & \text{otherwise} \end{cases} \tag{2.14}$$

3. Updating the cluster centers Z, by fixing the partition matrix U. With that, the objective function is minimized with:

$$z_{lj} = \frac{\sum_{i=1}^{n} u_{il} x_{ij}}{\sum_{i=1}^{n} u_{il}} l = 1, 2, ..., k, j = 1, 2, ..., d. \tag{2.15}$$

After the iteration, it returns the partition matrix U and the cluster centers Z.

The iteration process continues until a pre-defined criteria is met, typically minimising the sum of the distances to achieve the objective function of the k-means. The result of each iteration of the algorithm is guaranteed and typically converges to a local optimum - a solution that is optimal within a neighbourhood of candidate solutions.

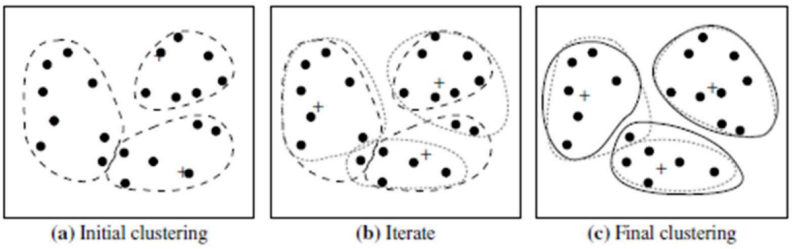


Figure 2.1: K-means iteration. Source: [14]

The k-means algorithm, as described, identifies clusters for a given value of k. There is no universal method for determining the optimal value of k, but there are several techniques for doing so, which will be explored in the following sections.

2.4.4 Optimal number of clusters

Finding the perfect number of clusters in a dataset is a key challenge in partitioning clustering, such as k-means clustering. This task involves the user specifying the number of clusters, denoted k, to be created.

Unfortunately, there's no one-size-fits-all solution to this puzzle. Determining the ideal number of clusters remains somewhat subjective, depending on the method of similarity measurement and the partitioning parameters used. Factors such as the nature of the data, domain-specific knowledge and the goals of the analysis also play an important role. Therefore, it often requires a combination of analytical techniques, experimentation and expert judgement to arrive at the most appropriate number of clusters for a given dataset.

2.4.4.1 Elbow method

The k-means algorithm has its drawbacks, such as the need to predefine the number of clusters and its sensitivity to the initial choice of cluster centres (centroids). Optimisation is essential to mitigate these problems. A commonly used method for cluster optimization is the Elbow method (Kodinariya and Makwana, 2013) [15] .

The Elbow method evaluates the total Within-Cluster Sum of Squares (WCSS) as a function of the number of clusters, in order to minimise the compactness of the clustering. The iteration starts with two clusters ($k = 2$) and increases by 1 in each iteration. Identify the "elbow point" on the graph, characterised by a noticeable slowdown in the rate of decrease of the WCSS. This inflection point indicates a decreasing marginal benefit in terms of explaining variance with the inclusion of additional clusters, and thus suggests an optimal number of clusters for the dataset.

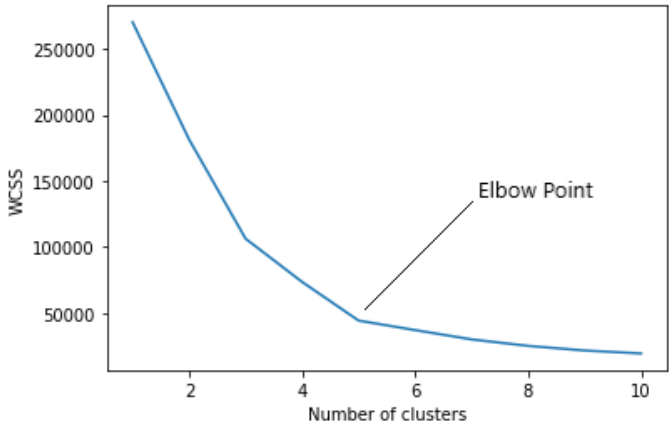


Figure 2.2: Elbow method example (source: [16])

As the number of clusters gradually increases, the distance to individual data points decreases, resulting in a decrease in the elbow metric until it converges to zero at k equal to the total number of data points. Consequently, plotting the mean distance to the centroid against the varying values of k reveals a distinct stabilisation point, commonly referred to as the elbow point, which serves as the key indicator for determining the optimal value of k.

2.4.4.2 Average Silhouette Width method

Along with Elbow Method, the Average Silhouette Width (ASW) method is one of the most popular index. Arbelaitz et al. (2012) [17] conducted an extensive study where in the ASW method demonstrated notably strong performance across the board.

Given a set of n data points and $X = (x_1, x_2, ..., x_n)$ from a space χ, d being a dissimilarity distance of χ. Knowing that clusters are partition, it can be written by labels $l(1), ..., l(n) \in \mathbb{N}_k = 1, ..., k$ where $l(i) = r \Leftrightarrow x_i \in C_r$, $I \in \mathbb{N}_n$. In addiction, let's denote clusters sizes $n_r = \sum_{i=1}^{n} \mathbb{I}(l(i) = r), r \in \mathbb{N}_k$.

Thus, the silhouette width for an observation $x_i \in \mathbf{X}$ is denoted by:

$$s_i(C, d) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{2.16}$$

where

$$a(i) = \frac{1}{n_{l(i)} - 1} \sum_{l(i)=l(j), i \neq j} d(x_i, x_j)$$

and

$$b(i) = \min_{r \neq l(i)} \frac{1}{n_r} \sum_{l(j)=r} d(x_i, x_j)$$

in case that $n_r > 1$ for $l(i) = r$. Otherwise, $s_i(C, d) = 0$. $a(i)$ represents the average distance from x_i to other points within the cluster it belongs to, while $b(i)$ represents the average distance from x_i to points within the nearest cluster it doesn't belong to.

The ASW of a cluster C is:

$$\bar{S}(C, d) = \sum_{i=1}^{n} s_i(C, d) \tag{2.17}$$

A high value of $s_i(C, d)$ indicates that $b(i)$ significantly exceeds $a(i)$, suggesting that x_i is significantly closer to the data points within its own cluster than to those in the neighboring cluster. The aim of clustering is to obtain clusters that are both internally homogeneous and well separated from each other. Therefore, larger values of s_i and average silhouette width indicate better clustering quality. Therefore, an optimal clustering solution should aim to maximize $\bar{S}$, as shown in the following equation:

$$\bar{S}(C^*, d) = \max_{C} \bar{S}(C, d) \tag{2.18}$$

2.5 Classical Linear Model

Before we dive into GLM, let's revisit the linear regression, both in its univariate (one-dimensional) and multivariate (multi-dimensional) forms. At its core, the concept behind multivariate linear regression is very similar to its simpler one-dimensional counterpart. Consider a data set with n data points, each point consists of a pair of values: x_i (representing an input or independent variable) and y_i (representing an output or dependent variable). In linear modelling, the relationship between the variables is linear, however this underlying relationship has a layer of error, symbolised by ϵ.

$$Y = ax + b + \epsilon \tag{2.19}$$

The error ϵ is assumed to be iid, with a Normal distribution with mean 0 and variance σ^2.

The problem of linear modelling is then reduced to finding the values of a and b. The classical way to solve this problem is by least squares regression, estimates for a and b by to minimising the following equation:

$$L(a, b) = \sum_{i=1}^{n} \left(y_i - ax_i - b\right)^2 \tag{2.20}$$

with the values of parameters that minimise $L(a, b)$ are

$$\hat{a} = \frac{\sum x_i y_i - \frac{\sum x_i \sum y_i}{n}}{\sum x_i^2 - \frac{(\sum x_i)^2}{n}} \text{ and } \hat{b} = \frac{\sum y_i - a^* \sum x_i}{n} \tag{2.21}$$

Considering now the multivariate linear model, instead of having a single dependent variable, we have k different variables so that the equation (2.19) results in:

$$Y = a_0 + a_1 x_1 + \ldots + a_k x_k + \epsilon \tag{2.22}$$

In order to minimise the square loss function, the best fit function is in this case:

$$L(a_0, a_1, ..., a_k) = \sum_{i=1}^{n} = \left(y_i - a_0 - \sum_{r=1}^{k} a_r x_{i,r} \right)^2 \tag{2.23}$$

where is the solution to minimization can be expressed in matrix term:

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \tag{2.24}$$

where $\mathbf{Y} = \{y_1, ..., y_n\}^T$ and $\boldsymbol{\beta} = \{a_0, a_1, ..., a_n\}^T$ are the vectors of observations and parameters, respectively. Also $\mathbf{X}$ is a matrix where each row is represent as $x_i = [1, x_{i1}, ..., x_{ik}]$.

2.6 Generalised Linear Model

The use of GLM for rating factor selection has been a major triumph in actuarial mathematics. While actuaries have always incorporated factor selection into pricing strategies, relying on methods such as multi-way analysis and minimum bias techniques, GLMs have solidified this process through large-scale analysis and robust diagnostic tools [18].

The power of GLMs lies in the ability to bridge the gap between simple linear regression and the multiplicative distributions that dominate actuarial analysis. This advance allows for more flexible relationships between rating factors and the outcome variable than the simple linear summation used in traditional methods. In essence, GLMs provide actuaries with a more powerful and adaptable framework for modelling risk and setting appropriate insurance prices.

GLMs share similarities with the traditional linear model, but include several important extensions. Unlike the linear model, which uses a simple linear combination of inputs to describe the expected value of the dependent variable, GLMs generalise this approach.

$$g^{-1}(\eta) = \mathbb{E}[Y] = a_0 + a_1 X_1 + ... + a_n X_n \tag{2.25}$$

where Y is the response variable and X_i $(i = 1, 2, ..., n)$ are the predictors. GLM extend linear models by introducing a link function, $g(.)$. This function transforms the response variable to address limitations such as ensuring positivity or modelling probabilities between 0 and 1. This allows to write $\mathbb{E}[Y] = g^{-1}(\eta)$. Unlike linear models, which assume a linear relationship between the response and predictors, GLMs allow a non-linear relationship through this transformation.

Unlike Gaussian linear models, which assume Normal errors, GLMs allow for a wider range of error structures from the Exponential Family:

$$f(\theta, y, \phi) = exp\left[\frac{\theta y - b(\theta)}{a(\phi)} + c(y, \phi) \right] \tag{2.26}$$

where θ and ϕ are scalar parameters and $a(.), b(.)$ and $c(.)$ are known real functions. θ is the scaling factor for the variance and ϕ is the dispersion parameter.

By specifying these parameters, we can obtain well-known distributions such as Poisson and Negative Binomial in the discrete case, or Gamma for continuous variables.

2.6.1 Deviance Tests

The distance between model predictions and actual observations is measured by model deviance. It generalises the concept of the sum of squared errors used in least squares regression. We can first consider the variance in terms of the scaled deviance, which is the difference between the log likelihood of the saturated model M_{full} (the model that fits the data perfectly) and the log likelihood of the fitted model multiplied by 2.

$$D_M^* = 2\left[L_{full} - L_{\hat{\mu}}\right] \tag{2.27}$$

In addition, there is an unscaled deviance of the model M, represented by

$$D_M = 2\hat{\phi}\left[L_{full} - L_{\hat{\mu}}\right] \tag{2.28}$$

where $\hat{\phi}$ is an estimated dispersion parameter. Also, considering two different models, M_1 and M_2, the difference in deviance between the models follows an asymptotic chi-square distribution. The degrees of freedom for this distribution are equal to the difference (Δp) in the number of estimated parameters between the two models.

$$D_{M_1}^* - D_{M_2}^* \overset{a}{\sim} \chi^2_{\Delta p}. \tag{2.29}$$

2.7 Decision Trees

Decision trees have gained popularity in various applications due to their ease of explanation and simplicity. These models provide a straightforward approach to predicting the response variable Y_i, making them a choice for many tasks in the insurance industry.

In the context of decision trees, the models use a tree-like structure to make decisions based on the characteristics of the input data x_{ij}. These models can be applied to both regression and classification problems. In the context of this work, where the response variable is numerical, we will dive into regression problems

For regression problems, where the response variable Y_i is quantitative in nature, the decision trees are designated as regression trees. The goal of a regression tree is to partition the input space in such a way that the variability of the response variable within each leaf node is minimised. This is achieved by recursively splitting the data into two child nodes, with each split being made to minimise some measure of impurity, such as mean squared error.

The advantage of using regression trees is their interpretability and their ability to capture non-linear relationships between the input features and the response variable. By examining the tree structure and the decision rules at each node, users can gain insight into the underlying patterns in the data and the factors driving the predictions [19].

Consider a collection of observations $y_1, y_2, \ldots, y_n$ of the response variable Y_i. Each observed value y_i (for $i = 1, 2, \ldots, n$) depends on the explanatory variables $x_1, x_2, \ldots, x_p$. To model this dependence, we divide the predictor space - which includes all possible values of $x_1, X_2, \ldots, x_p$ - into J distinct and non-overlapping regions, $R_1, R_2, \ldots, R_J$. For each observation that falls into a region R_j, we predict the same value, which is the mean of the response values for the training observations within R_j.

Nevertheless, to build a decision tree, we divide the predictor space into J high-dimensional rectangles or boxes to facilitate interpretation of the resulting predictive model. We evaluate all possible split points for each of the p predictors $X_1, X_2, ..., X_p$ and select the predictor and split point that minimize the Residual Sum of Squares (RSS) to construct a tree with the lowest error[19].

In essence, the goal is to find boxes $R_1, R_2, ..., R_j$ that minimizes the RSS:

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \hat{y}_{R_j} \right)^2 \tag{2.30}$$

where $\hat{y}_{R_j}$ is the mean response for the training data set within the j_{th} box.

Considering all possible partitions of the feature space into J boxes is computationally infeasible. Therefore, a top-down greedy approach called recursive binary splitting is used. This is a top-down process that starts at the root of the tree, where all observations belong to a single region, and then recursively splits the predictor space by creating two new branches at each step. This process continues, with each split defining two new regions further down the tree, until the desired level of complexity is reached.

2.7.1 Regression Tree Example

For this example, we will use a fictitious randomly generated dataset to illustrate the process of building a decision tree. The data consists of 1000 observations of a single explanatory variable called DriverAge, which is used to estimate the target variable Claim Amount, also generated at random.

Although several explanatory variables are used in the full case study to enrich the predictive modelling, in this specific example we will only focus on the $DriverAge$ variable. The aim of this simplification is to make it easier to understand the process of building and interpreting decision trees, without losing sight of the fundamental concepts.

For this modelling example, we will define $max\_depth = 3$ and $max\_leafs = 4$.

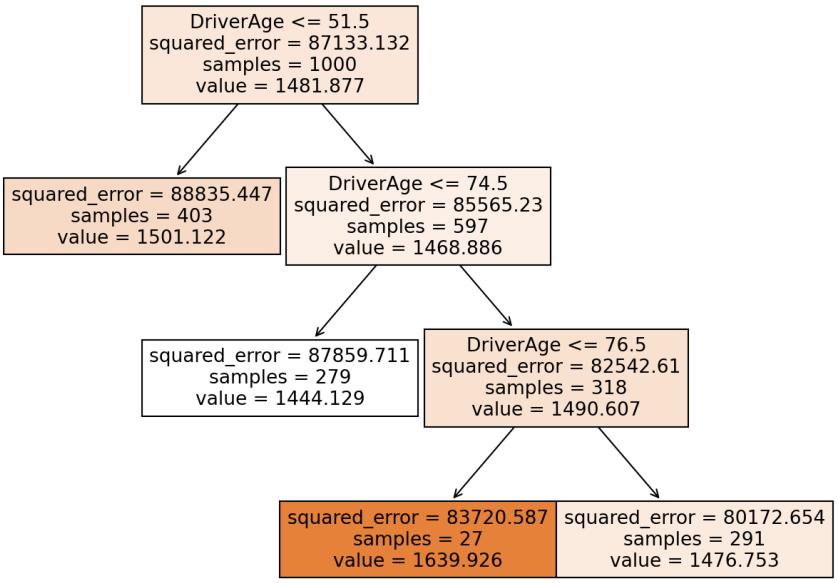


Figure 2.3: Regression Tree Example (authors source.)

In this decision tree regression, the algorithm evaluates each possible split on each feature to find the one that maximises the reduction in squared error, which is a measure of how far the predicted values are from the actual target values. The tree structure starts with a root node where the initial decision is made: whether $DriverAge$ is less than or equal to 51.5. This node contains 1000 samples and predicts a $ClaimAmount$ of 1481.88. The first split in the decision tree occurs at $DriverAge \leq 51.5$ because this split minimises the square error of all possible splits.

If the condition is true, the tree follows the left branch to a node containing 403 samples with a predicted value of 1501.12. If false, the tree moves to the right branch where it continues to evaluate $DriverAge$ with respect to 74.5. This process of splitting continues until leaf nodes are reached, in this case the maximum number of leaves is defined as 4, representing final predictions for specific subsets of the data.

At each node, the model calculates the squared error to measure the accuracy of the prediction. A lower squared error indicates a more accurate prediction. The tree aims to minimise this error by dividing the data into smaller, more homogeneous groups. The number of samples at each node reflects the size of the data in that subset, decreasing as the tree splits. The predicted values at the nodes represent the average target variable for the corresponding samples. As the tree splits, the predictions adjust based on the characteristics of the smaller groups. In this case, the tree only uses $CarAge$ to split the data, aiming for the greatest reduction in error.

When making a prediction for $ClaimAmount$, the tree follows the decision path based on $DriverAge$. For example, a 55 year old driver would first be compared to a 51.5 year old driver. If the condition is false, the tree moves to the right branch and then evaluates

against 74.5. This process continues until a leaf node is reached, which provides the final prediction for $ClaimAmount$.

2.7.2 Regression Trees Implementation Drawback

Decision trees offers a clear advantage in terms of interpretability. Their structure allows users to visualise decision points at each node, facilitating informed decisions based on the logic of the model. In particular, they excel at handling non-linear relationships and multiple data types (numeric and categorical) without data transformation, making them powerful tools for large datasets.

However, decision trees are prone to overfitting, especially with deep structures. These complex models can closely mimic the training data, capturing noise as well as underlying patterns. Techniques such as hyperparameter tuning can help mitigate this problem. In addition, the rectangular splits in the feature space can lead to discontinuities in the predictions, limiting their accuracy for continuous variables.

Due to these limitations, we will explore ensemble methods that build on decision trees. Ensemble learning techniques, which combine multiple models to improve performance and reduce overfitting, address many of these drawbacks and provide a robust framework for building effective predictive models in our case study.

2.8 Gradient Boosting

First introduced by Friedman (2001) [20], GBM is a powerful ensemble technique designed specifically for decision trees. It combines multiple, weaker tree models into a single, more robust predictor. Ensemble methods generally use the collective wisdom of multiple learners to achieve superior performance compared to individual models. This is particularly beneficial for decision trees, which can suffer from both low bias and high variance . Combining multiple trees into an ensemble can address both of these issues.

The GBM is a versatile algorithm, applicable to both classification and regression tasks. They are praised for their efficiency and incorporate mechanisms to reduce overfitting. These mechanisms include randomisation and regularisation techniques that decorrelate the individual trees within the ensemble.

The application of GBM, especially in insurance, has grown significantly in recent years. In this context, our goal is to use GBM to learn the function $f(x)$ that accurately maps input features (X) to target variables (Y)[20].

In the following, we will describe the steps of the algorithm in detail and explain the role of the most important hyper-parameters. Taking D as the training set, $D = (y_i, x_i); i = 1...N$ and N the number of samples in our training set.

Initialize model with a constant value, which is the base predictor.

$$F_0(x) = argmin_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$$

where y_i refers to the observed values, γ refers to the predicted values and the sum means that we add up one differentiaLoss bFunction for each observed value. So the $argmin$ over γ means we need to find the predicted values that minimizes this sum.

Next, in order to form the model it is implemented M decision trees, $m = 1, ..., M$.

Firstly, it is computed

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i)}{\partial F(Xx_i)}\right]_{F(x)=F_{m-1}(x)} \text{for } i = 1, ..., n$$

where i is the sample number and m is the tree built. Partial residuals are an intermediate metric spread across all the iterations of the algorithm from 1 to M, and they will also depend on the loss function of choice.

Next the algorithm fits a regression tree to the r_{im} values and create terminal regions R_{jm}, for $j = 1, ..., J_m$.

Next for $j = 1, ...J_m$ it is computed

$$\gamma_{jm} = argmin_\gamma \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$$

Lets take note that this step is very similar from the step where it is defined $F_0(x)$ but now it has the previous prediction into account.

And it is updated $F_m(x)$ by making a new prediction for each sample

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_m \mathbb{I}(x \in R_{jm})$$

A key aspect of GBM is how each new tree builds on the previous one. This sequential refinement is controlled by a parameter called the learning rate ν, which is typically set between 0 and 1. A smaller learning rate reduces the influence of each individual tree on the final prediction. This may seem counterintuitive, but it helps prevent overfitting and improves overall accuracy in the long run.

As highlighted earlier, GBM shares similarities with GLMs in its goal of minimising a loss function. However, it introduces additional hyperparameters, such as the maximum depth of individual trees. This parameter controls the "weakness" of each learner, but also its ability to capture complex interactions between features. Finding the right balance between these factors is crucial during the model tuning process.

While the interpretability of a GBM model may be lower than that of a single decision tree (a common drawback of ensemble methods), techniques such as partial dependence plots and feature importance plots can still provide valuable insights. These tools can help understand how individual features contribute to the final prediction, even within the complex ensemble structure.

2.8.1 Cross-validation

Cross-validation is a cornerstone technique for model selection in machine learning. It is based on the hold-out method, but makes more efficient use of a single data set. This

technique dynamically splits the data into subsets, allowing each subset to play the role of both training and testing (selection) sets at different times. As a result, cross-validation uses all data points for training.

This approach is crucial because the test set is not used to influence model configuration or hyperparameter tuning. Doing so, allows the model to learn from unseen data, however subtly, leading to overly optimistic performance estimates. To prevent this, cross-validation uses a separate part of the data for validation, typically separated from the original training set. This provides a more robust assessment of the generalisability of the final model.

K-fold cross-validation is a technique that estimates prediction error by dividing the dataset into K distinct subsets, where common choices for K are 5 and 10. In this method, each subset $k = 1, \ldots, K$ is sequentially excluded from the dataset and the model is trained on the remaining $K - 1$ subsets. The excluded subset is then used as the test set. This process is repeated until each subset has been used as a test set. The prediction error is then calculated using the chosen loss function over all K subsets.

$$CV^{(K)} = \frac{1}{K} \sum_{k=1}^{K} L_k \tag{2.31}$$

This method allows the full use of all available data, rather than dividing the data set into test and training data sets. By running the train/test folds several times, a best estimate of the model performance can be obtained.

A key advantage of cross-validation is its ability to use all available data. By dynamically partitioning the data into these folds and iteratively using them for training and testing, cross-validation ensures that all data points are included in the model building process. This approach effectively maximises the use of available data, resulting in a more robust estimate of model performance.

3

Case Study and data treatment

The dataset used in this study aims to capture the complexity and scale of the data typically encountered in insurance companies. To this end, we have selected a substantial policy portfolio that enables the extraction of statistically significant patterns. This data comprises a variety of features that can be used to forecast the behavior of typical motor response variables. The dataset used in this work is from an automobile Third Party Liability (TPL) portfolio of an insurance company.

Our initial data cleaning process targeted inconsistencies that could throw off our models, and as part of this, we removed one policy with exactly 16 claims, identified as an error.

3.1 Characterization of the data set

The dataset used in this analysis, while real, is outdated and serves primarily as an illustrative example of the methods studied. The claims amounts presented in this work are no longer current and should not be interpreted as accurate or reflective of current conditions. In our analysis, we considered the following features describing each policy, labelled as $ncontract$:

- $Exposure$: Risk Exposure of each policy during the year;
- $Zone$: A column based on the *County*;
- $Power$: Vehicle power which is classified into categorical levels ranging from 4 to 15;
- $CarAge$: Vehicle age measured in years;
- $DriverAge$: Numerical variable representing the age of the usual driver;
- $Brand$: Brand of the insured vehicle. Classification in categorical levels, ranging from 1 to 11;
- $Fuel$: Car gas type, either Diesel (D) or Gasoline (E);

- $PopDensity$: Numerical variable with population density as inhabitants per km^2 in the driver region;
- $County$: County of residence from the usual driver Classification in categorical levels, raging from -1 to 13;
- $ClaimAmount$: The cost of each claim in Euros (€);
- $ClaimNb$: Number of claims for the observed exposure period for each policy.

Given the raw data available and in order to arrange the data, it was built two files where it is possible to model the frequency and severity of claims. The dataset consists of two dataframes, $baseFREQ$ and $baseSEV$, which are described as follows:

- $baseFREQ$: An automobile TPL, containing 49,999 policies, includes a range of risk features along with the claim counts for each policy based on their exposure during the observation period.
- $baseSEV$: This dataframe contains 1,908 claims derived from the frequencies reported in the $baseFREQ$ dataframe. All claim costs are greater than €0.

The $ncontract$ column in each dataframe identifies the policy number. If a contract has more than one row in $baseSEV$ dataset, this indicates that the same contract in the $baseFREQ$ dataset has $ClaimNb > 1$.

3.1.1 Claim Counts analysis

As mentioned earlier, with the dataframes prepared, it is now possible to perform a predictive analysis of the model's response variables. $ClaimNb$ is a variable that reflects the number of claims per policy.

An important observation is that the majority of policies have no claims, as shown in figure 3.1. In this portfolio we have 47,510 policies (95.02%) with no reported claims. In contrast, 2,313 policies reported one claim, 162 policies had two claims, 12 policies had three claims and only 2 policies reported four claims. This distribution clearly shows a strong bias towards policies with no claims, which is common in TPL portfolios.

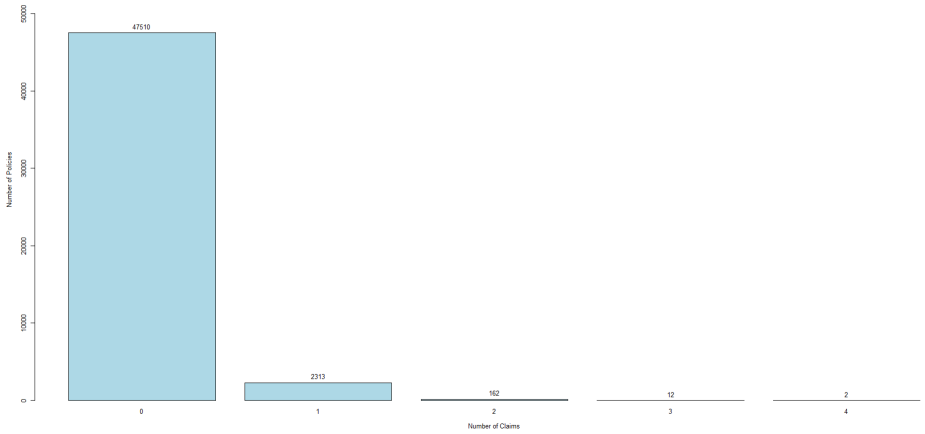


Figure 3.1: Barplot of $ClaimNb$ variable

This allows to calculate and analyse preliminary statistics of the portfolio, where we can observe the values shown in the following table:

Mean	Variance	Standard Deviation	Skewness	Kurtosis
0.05362	0.05914	0.2432	4.9743	31.8464

Table 3.1: Preliminary statistics analysis of ClaimNb

Of the 49,999 policies considered in this study, we observe an average claim frequency of 0.05362 with a variance of 0.05914. The skewness value of 4.9743 suggests that the distribution of claims is highly asymmetric to the right. This means that there are many cases with a low number of claims and a few cases with a very high number of claims. The kurtosis value of 31.8464 indicates that the distribution of the number of claims is extremely peaked, resulting in a very narrow distribution with most of the responses on the left, in this case for 0 claims reported. To get a more comprehensive view, we also analysed the number of claims using exposure as a weighting factor. The results of this analysis are shown in the table below:

Mean	Variance	Standard Deviation	Skewness	Kurtosis
0.10143	0.3118	0.55839	116.4651	19,100.16

Table 3.2: Preliminary statistics analysis of ClaimNb weighted by Exposure

This exposure-adjusted analysis provides a more standardised view of claims frequency, taking into account different policy exposures and providing a more accurate representation of risk across the portfolio. When the number of claims is weighted by exposure, we observe that the claims frequency approximately doubles.

3.1.2 Claim Severity analysis

The *ClaimAmount* variable will serve as another response variable for our models, where it will adjust the severity models. Initial statistical analysis revealed that the mean claim amount for each claim is €1,715.51, with a standard deviation of €3,449.71. The maximum reported Claim Amount is €75,000, while the minimum value is €0.01. To further analyze the distribution and quantiles of the Claim Amount, we examined the 50th, 90th, 95th, 97.5th, and 99th percentiles. This analysis provides a more comprehensive understanding of the claim amount distribution, particularly focusing on the upper tail of the distribution, which is crucial for assessing the impact of large claims on the overall portfolio.

50%	90%	95%	97.5%	99%
€1,172.0	€2,936.6	€5,044.4	€7,784.4	€16,238.2

Table 3.3: *ClaimAmount* Quantiles

Based on the quantile analysis, we can observe that 90% of the reported claims are below about €3,000. This observation suggests a significant influence of severe claims on the overall distribution. This finding is further supported by the boxplot shown in Figure 3.2. The box plot visually shows a significant dispersion of claims above the 95% quantile, indicating the presence of outliers.

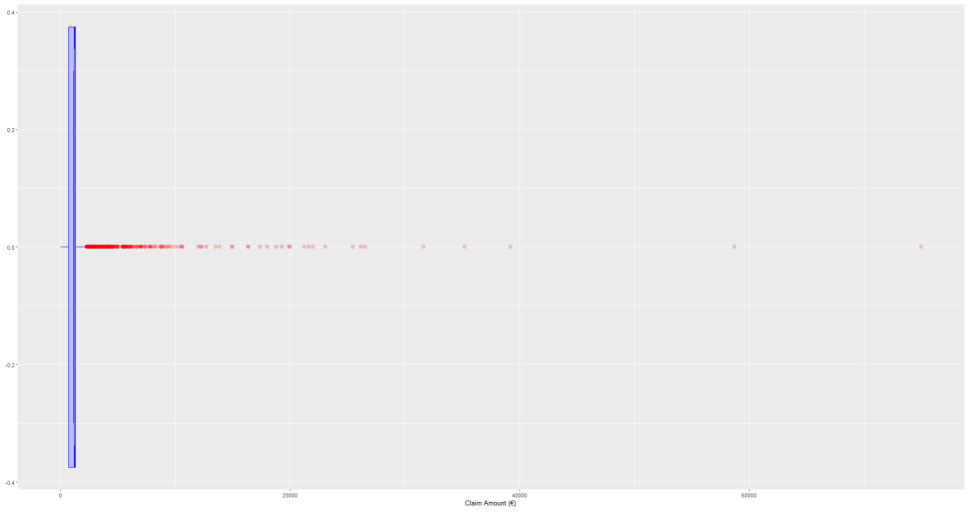


Figure 3.2: Boxplot of *ClaimAmount*

To gain a better understanding of common claims and their distribution, we focused our analysis on claims below €6,000. This approach gives a clearer picture of the typical claim pattern. By examining this subset of claims, we found that a significant proportion of these common claims fall within the €1,500 to €1,750 range. This concentration provides valuable insights into the most common claim amounts for non-catastrophic incidents. In

the upcoming distribution fitting sections, we will further refine our analysis by splitting the dataset into two distinct categories.

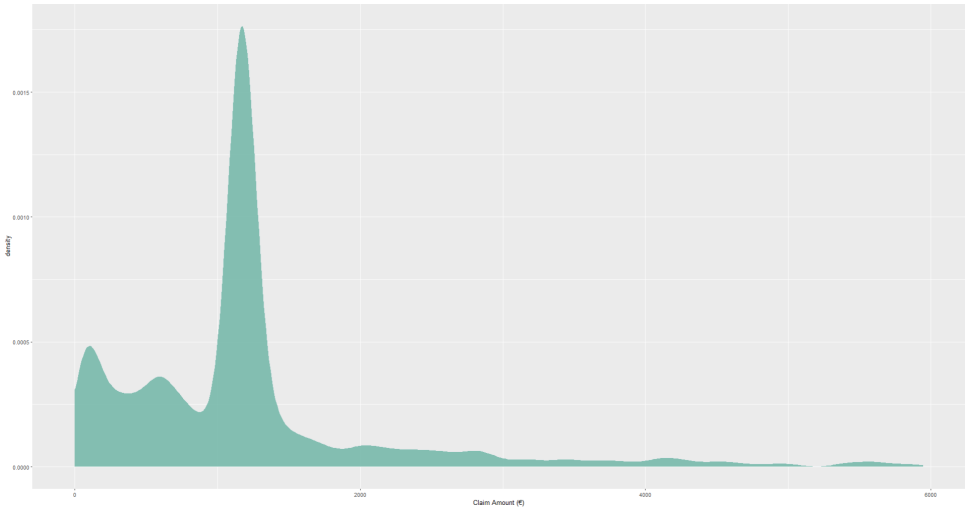


Figure 3.3: Density plot of $ClaimAmount$

3.2 Exploratory Data Analysis

3.2.1 Driver Age

The variable $DriverAge$ plays a crucial role in risk modelling in motor insurance, being highly significant in both GLM and GBM models. Driver age is often correlated with the frequency and severity of claims, as it reflects both driving experience and risk propensity, factors that are fundamental to the correct pricing of insurance policies.

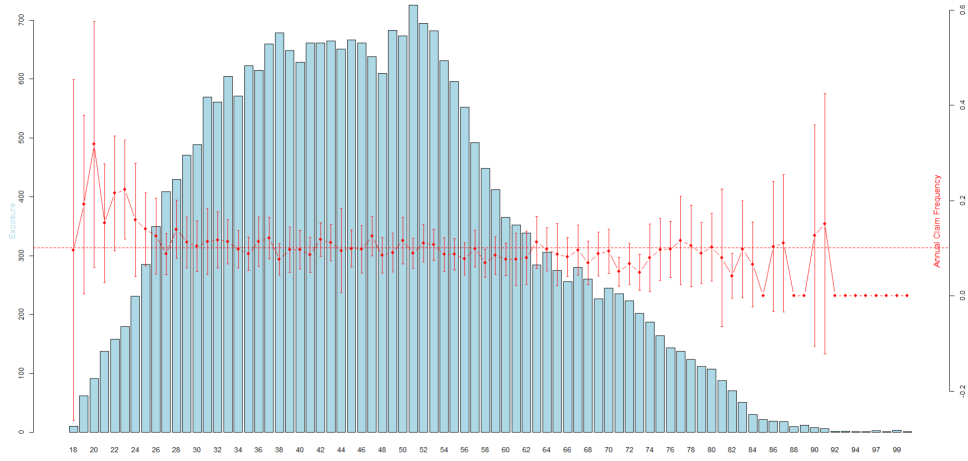
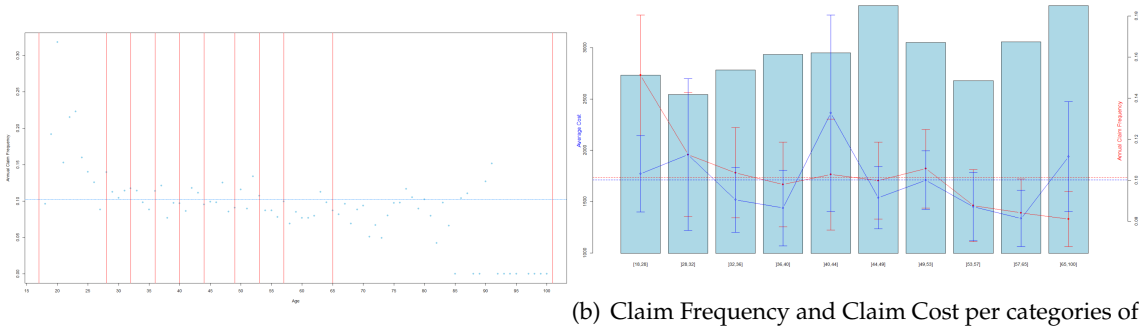


Figure 3.4: Claim Frequency vs Exposure per $DriverAge$

The bar plot clearly shows that drivers aged up to 25 years have a significantly higher claim probability compared to older drivers. However, it is important to note that this age group is not well represented in the dataset, which can be observed from the exposure bars.

Firstly, to better analyse the impact of driver age on claims, the numerical variable can be transformed into a categorical variable using 10% quantiles. This segmentation technique allows the driver population to be divided into identical exposure groups, making it easier to identify risk patterns. However, we do not take claim frequencies into account when applying the quantile-based age segmentation. Our aim is to further analyse the scatter plot, where we can assess whether the age quantile split has been effectively applied. In addition, categorising age can help to identify specific age groups with a higher or lower risk of claims, while reducing the impact of outliers by smoothing out extreme variations that could skew the results. The driver age variable was therefore divided into 10 groups: $[18, 28]$, $]28, 32]$, $]32, 36]$, $]36, 40]$, $]40, 44]$, $]44, 49]$, $]49, 53]$, $]53, 57]$, $]57, 65]$, $]65, 100]$.



(a) Average Claim Frequency per DriverAge

(b) Claim Frequency and Claim Cost per categories of $DriverAge$

Figure 3.5: First categorical approach of $DriverAge$ feature

Looking first at the dot plot, we can see that the analysis of the driver's age into categories fits well for older ages. However, the first interval between 18 and 28 is exaggerated, as there is an observable difference in frequency between 18 to 24 and 25 to 28. This is reflected in the concrete values, since if we analyse the average frequency of the ages between 18 and 28, it is about 0.1485. However, if we look at the population aged between 18 and 24, the empirical claim frequency increases to 0.2145, while for those aged from 25 to 28 is 0.1231. On the other hand, the categories $]32, 36]$ and $]36, 40]$ are quite similar, both in terms of claim frequency and claim costs. In the first interval we have a claim frequency of 0.0975 and an average cost of €1,575, while in the interval $]36, 40]$ we have a claim frequency of 0.0968 and an average cost of €1,436. We will therefore combine these two categories. For the remaining categories, we will leave them separated by quantile, as there are differences in claim frequency and cost between them.

Categories	Nr Policies	Nr of claims	Average Cost	Claim Frequency
[18, 24]	1,583	137	€1,945	0.2145
]24, 28]	2,926	157	€1,589	0.1231
]28, 32]	4,305	229	€1,957	0.1170
]32, 40]	10,051	511	€1,477	0.1030
]40, 44]	5,009	270	€2,362	0.1032
]44, 49]	6,067	325	€1,537	0.1007
]49, 53]	5,029	298	€1,744	0.1073
]53, 57]	4,421	223	€1,449	0.0906
]57, 65]	5,210	259	€1,335	0.0863
]65, 100]	5,399	288	€1,937	0.0815

Table 3.4: $DriverAge$ intervals analysis

This table provides a clearer insight into the claims patterns of different age groups. Policyholders aged [18, 24] have a significantly higher claim frequency than other age groups, indicating a higher level of risk that is reflected in the tariff premiums and the coefficients generated by the GLM, particularly in the frequency model.

Even within a TPL portfolio, younger drivers have higher average costs compared to the rest of the portfolio. In this particular portfolio, drivers aged 41 to 44 have the highest claim costs associated with severe accidents.

The higher risk profile of younger drivers is largely due to their limited driving experience. In contrast, drivers over the age of 53 have a lower annual claim frequency, indicating a lower risk profile. This suggests that older drivers are more likely to adopt a defensive driving style than their younger counterparts.

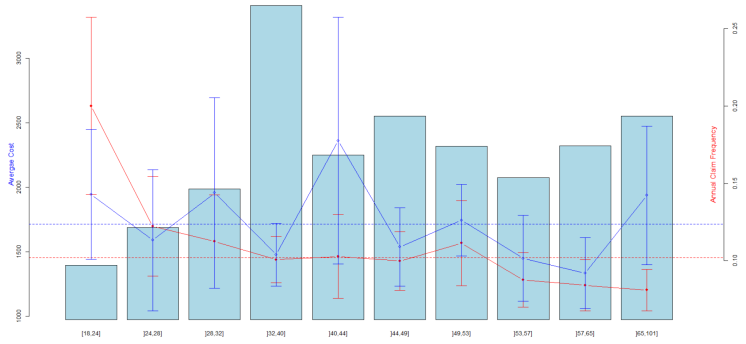


Figure 3.6: Claim Frequency and Claim Cost per $DriverAge$ categories

3.2.2 Car Age

The $CarAge$, a numerical variable ranging from 0 to 100 years, was transformed into a categorical variable using the same approach as the $DriverAge$ variable. The analysis showed that 92.5% of the portfolio consists of vehicles less than 15 years old. For vehicles

up to 16 years old, the annual claim frequency remains relatively stable. However, for older vehicles, the data shows a greater dispersion and a tendency for the claim frequency to increase, probably due to the lower number of policies in this age group.

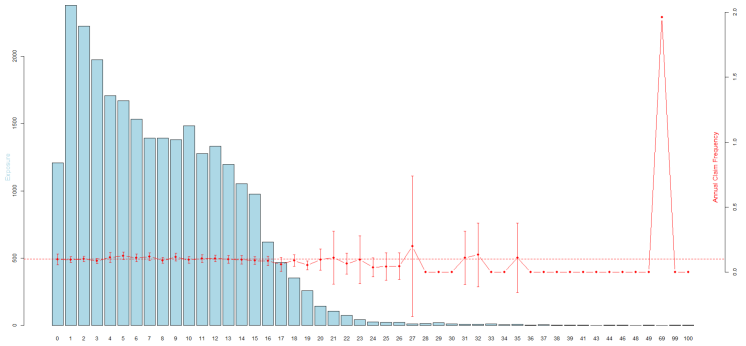
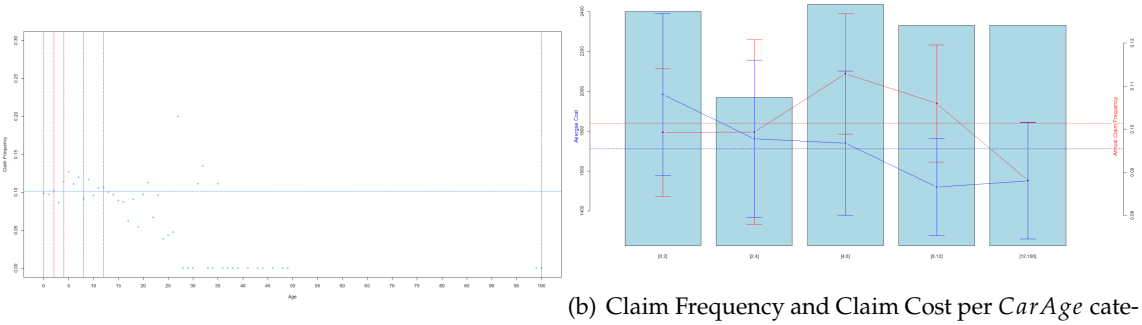


Figure 3.7: Claim Frequency and Claim Cost per $CarAge$

First, the $CarAge$ variable was divided into 5 groups using 20% quantiles to identify specific risk patterns for each age group. We started by using 20% quantiles rather than 10% quantiles due to a large proportion of the data was below the age of 15, which would have resulted in many categories having only a 1-year interval. Dividing by quantiles ensures that each group contains approximately the same number of contracts, making it easier to compare the segments. Given this, the categories formed are: $[0, 2]$, $]2, 4]$, $]4, 8]$, $]8, 12]$, $]12, 100]$.



(a) Average Claim Frequency per $CarAge$

(b) Claim Frequency and Claim Cost per $CarAge$ categories

Figure 3.8: categorical approach of $CarAge$ variable

Despite the overall differences, some groups show similar behaviour and will be grouped together later in the GLM application. This is because frequency and severity are analysed separately. The patterns observed suggest that while newer cars have lower claim frequencies, their claims tend to be more expensive. Conversely, older cars have higher claim frequencies but lower average costs per claim.

3.2.3 Power of Vehicle

Vehicle power is a categorical variable in this dataframe with values ranging from 4 to 15. The figure (3.9) shows that the vast majority of exposures are concentrated at levels 4, 5, 6 and 7. At these levels there is not as much variation in either frequency or average cost. In the remaining categories, the increase in value fluctuations is noticeable, especially the very low average costs in classes 11, 12 and 13, although the frequency in the last two classes is above average.

Looking at the data for the other categories, a different pattern emerges. In these classes there is much greater variation in values. This is particularly the case in classes 11, 13 and 14, where the average cost is below the average, although the frequency in classes 13 and 14 is above the average. It is also important to consider the low frequency in some of the higher classes, such as 14 and 15. These classes are less represented in the dataset, which can affect the statistical analysis, making the data from these categories less reliable for generalised conclusions.

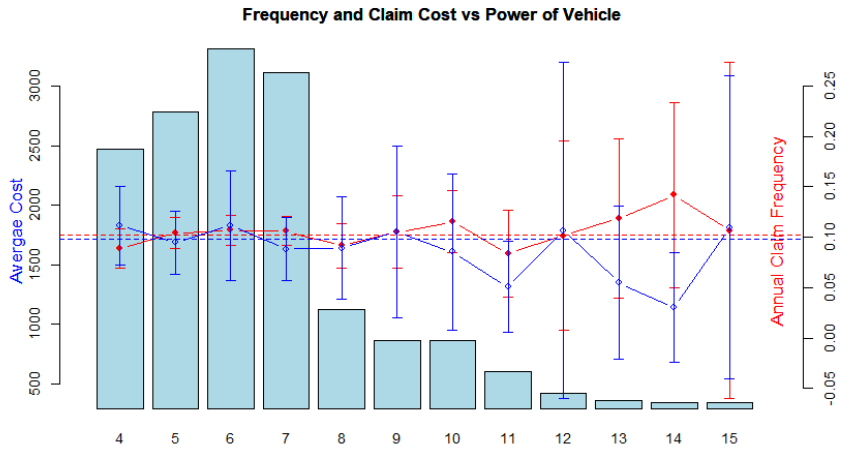


Figure 3.9: Average Claim Frequency and Claim Cost per *Power*

3.2.4 Fuel

Fuel is a variable in our dataset with two different categorical values, "D" and "E". Category "E" is the predominant fuel type in our portfolio. More importantly, it has a lower risk profile than category "D". This difference in risk is evident across a number of metrics. The average cost of claims also follows a similar pattern, with category "E" vehicles incurring lower costs than those in category "D". This trend in claim severity further reinforces the lower risk profile associated with category "E" vehicles.

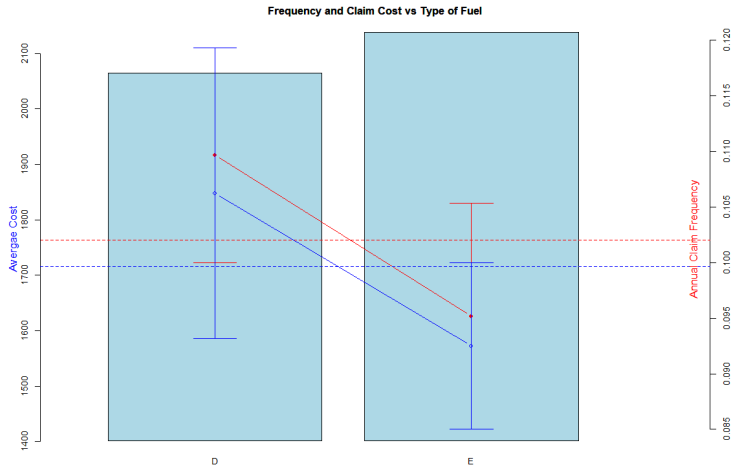


Figure 3.10: Average Claim Frequency and Claim Cost per *Fuel*

3.2.5 Brand

Brand is a categorical feature that represents the brand of the insured object. We have 11 categories with values ranging from 1 to 6 and 10 to 14. It can be seen that categories 1, 2 and 12 represent a larger part of the portfolio, and in these categories the range of confidence intervals is smaller due to the number of observations in each category, as expected. It is noteworthy that there are several categories with very similar values for both annual claim frequency and average cost, which would be expected to be further explored with category groupings in the GLM. Category 13 has the highest average cost and claim frequency, indicating a higher risk. However, this may be due to the small sample size of this category in the portfolio.

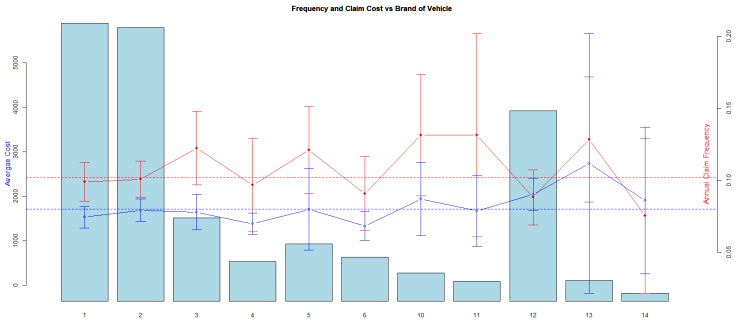


Figure 3.11: Average Claim Frequency and Claim Cost per *Brand*

3.2.6 Cluster Analysis

We understand that the variable $Zone$ is a categorical variable derived from the $County$ variable. By employing clustering techniques, we aim to create a more refined aggregation that considers not only the $County$ variable but also $PopDensity$.

Based on these considerations, we adopted a methodology in which each combination of the key variables $County$ and $PopDensity$ was analyzed. This included assessing the number of claims, risk exposure (defined as the sum of exposures) and claim cost for each combination. Additionally, we incorporated population density as a predictor variable within our clustering model. This approach enables us to form clusters in which the number of claims, exposure, claims cost and population density exhibit similar characteristics, thereby providing a more nuanced analysis of the data. With that, we have a dataframe with 326 observations as it is shown in the following figure.

	Key	NrClaims	Exposure	PopDensity	County	ClaimAmount
1	-1_11	15	46.302740	11	-1	18703.89
2	-1_21	0	2.440000	21	-1	0.00
3	-1_22	2	4.490000	22	-1	1128.12
4	-1_23	1	2.570000	23	-1	0.00
5	-1_24	30	66.863661	24	-1	48005.74
6	-1_25	1	4.265464	25	-1	1128.12
7	-1_26	5	7.890000	26	-1	4669.50
8	-1_31	5	14.998219	31	-1	10312.43
9	-1_41	6	3.050000	41	-1	5961.57
10	-1_42	2	1.480000	42	-1	16128.76

Figure 3.12: Cluster Analysis Dataframe

The Key column is a concatenation of the columns $County$ and $PopDensity$. The number of rows in the dataframe corresponds to the number of unique values in Key column. Upon examining the dataframe, we discovered that one key exhibited significantly higher values than all others. This outlier corresponds to the combination of $County$ 13 and $PopDensity$ 24. This specific combination accounts for a disproportionate 15% of reported number of claims and 13% of the total claim cost. Given its distinct nature, we anticipate that this combination will not form a cluster with any other Key.

In order to apply the K-means algorithm, two techniques were chosen to determine the optimal number of clusters: the Elbow Method and the Silhouette Method. Firstly, the Elbow Method is used to analyze the WCSS. This method aims to minimize the WCSS as the number of centroids increases. This can be observed:

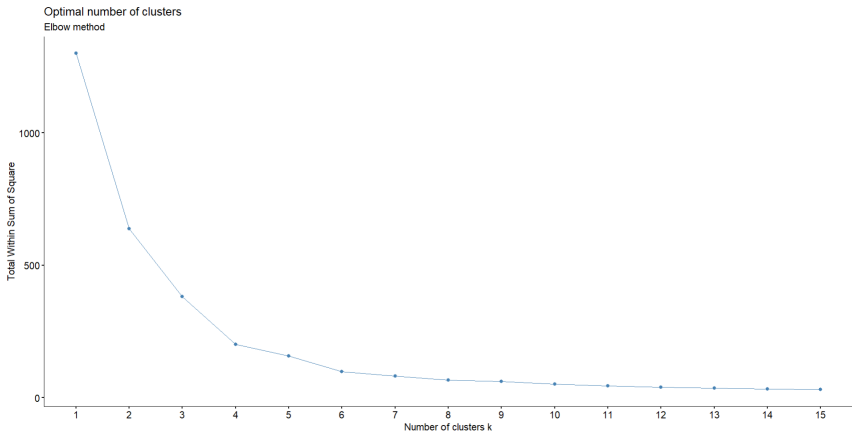


Figure 3.13: Elbow Method

It also can be displayed in a table, where we can analyze the differences between WCSS from the current centroid to the previous.

Clusters	**WCSS**	**ΔWCSS**
1	130,000,000	-
2	63,799,677	-66,200,323
3	38,172,924	-25,626,753
4	20,029,630	-18,143,294
5	15,690,989	-4,338,641
6	9,857,921	-5,833,068
7	8,080,792	-1,777,129
8	6,498,813	-1,581,979
9	6,075,129	-423,684
10	5,107,842	-967,287
11	4,440,878	-666,964
12	3,845,551	-595,327
13	3,614,035	-231,516
14	3,289,497	-324,538
15	3,014,212	-275,285

Table 3.5: WCSS and ΔWCSS for different clusters

By analysing the figure (3.13) and the table 3.5, we can conclude that, using the elbow method, the optimal number of clusters may occur around 5 and 8. The first few clusters (e.g., from 2 to 5 clusters) show a sharp decrease in WCSS, suggesting that adding more clusters in this range significantly improves the clustering. After 5 clusters, the rate of change in WCSS decreases. This leveling off is indicative of reaching a point of diminishing returns. After 5 clusters, the rate of change in WCSS decreases. This leveling off is indicative of reaching a point of diminishing returns. As the number of clusters increases beyond 8 clusters, the change in WCSS further decreases, indicating that adding more clusters is not providing much additional improvement. With that, we can conclude that clustering with 5 or 8 clusters provides a balance between capturing distinct groups

in the data while also avoiding overly complex and potentially less meaningful clustering.

On the other hand, the Silhouette score measures how well each data point fits within its cluster. By plotting the average Silhouette score for each potential number of clusters, one can identify the best value for k, as it will correspond to the highest Silhouette score.

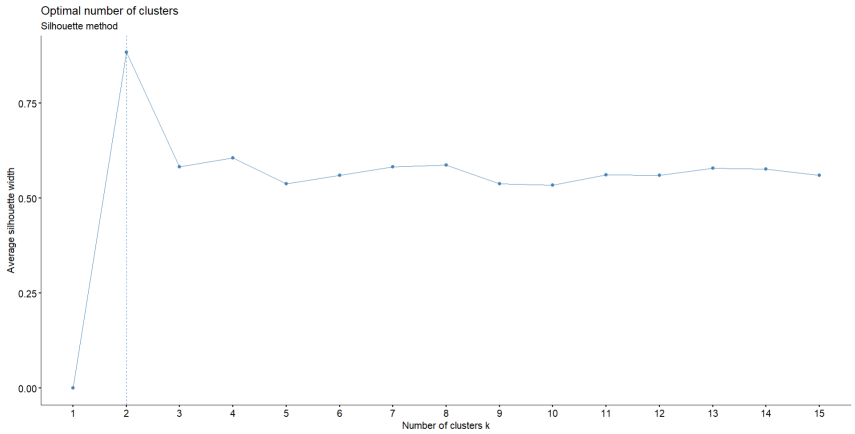


Figure 3.14: Silhouette Method

Based on the analysis of figure (3.14), the highest Silhoutte score corresponds to the 2 clusters. With that, and based on our data, we will prioritize the Elbow method by focusing on a more balanced representation of the data's structure, avoiding overly simplistic groupings. The elbow method suggests an optimal number of clusters around 5 or 8, providing a better capture of the data's complexity and nuances. Choosing a higher number of clusters based on the elbow method may offer more meaningful groupings and improved separation of the data points, despite a potential decrease in silhouette score.

We conducted an analysis using the k-means algorithm with the $kmeans$ function in R for different values of $k \in \{5, 6, 7, 8\}$. For each value of k, we visualized the centroids using the $fviz\_cluster$ function in R to evaluate the performance of the algorithm. This function provides a clear visual representation of the data points and the cluster centroids, making it easy to see how the data is grouped into different clusters.

First, the k-means algorithm was applied with $k = 5$. The outlier referred previously, was observed, represented as cluster 4 in figure (3.15). It was also noted that cluster 5 has a significant size, which indicates a substantial distance between the observations and the centroids. Clusters 1, 2, and 3 have the most observations, with dimensions of 118, 85, and 117, respectively.

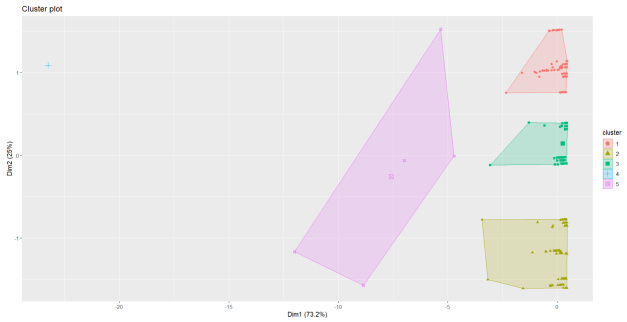


Figure 3.15: 5 clusters

Next, the k-means algorithm was applied with $k = 6$ centroids. The outlier was still present, now categorized in cluster 2. Cluster 5, previously represented as cluster 5 in the earlier analysis, transformed into cluster 1, as two observations were far from the centroid. Additionally, cluster 4, previously represented as cluster 3, decreased in the number of observations, shifting from 85 to 83.

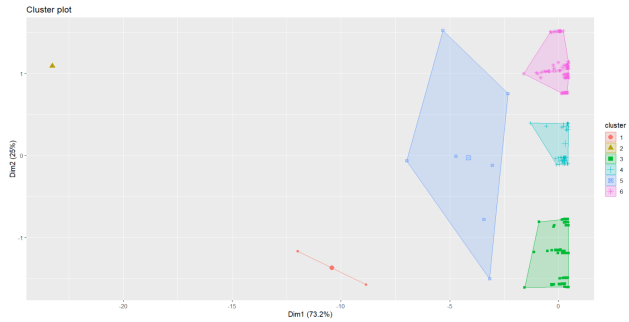


Figure 3.16: 6 clusters

Following that, the k-means algorithm was applied with $k = 7$ centroids, which led to a greater division of observations as expected, particularly in clusters 2 and 4. This division demonstrates the algorithm's ability to further refine clusters and separate data points more precisely.

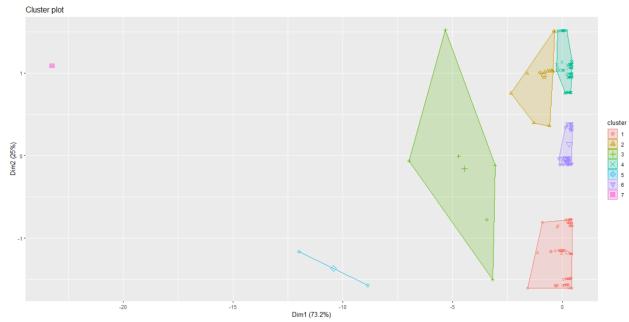


Figure 3.17: 7 clusters

Finally, when the algorithm was applied with $k = 8$ clusters, a greater division of the clusters was observed along with a shorter distance between each observation and its corresponding centroid. This suggests improved accuracy in clustering as the number of

clusters increased. With eight clusters, there is a more precise separation of data, which may facilitate the identification of specific patterns in this context.

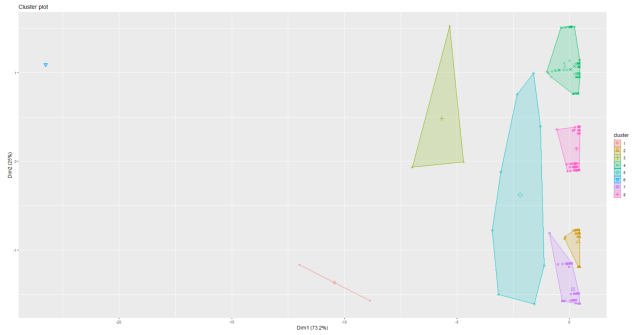


Figure 3.18: 8 clusters

In the end, we chose $k = 8$ because this configuration provided a more precise stratification of the data, revealing specific patterns and trends that may have been obscured with fewer clusters. The greater level of granularity allows us to capture as much information as possible, enabling a more nuanced understanding of the data. This detailed clustering serves as a valuable foundation for applying GLM and GBM models, as it helps us better segment the data and tailor the models to each specific group. By leveraging the insights gained from the k-means clustering, we can improve the accuracy and effectiveness of the subsequent models, ultimately leading to more reliable predictions.

Given that the cluster analysis was based on the variable Key, which represents the combination of $County$ and $Popdensity$, a dataframe was created to identify each Key with its respective cluster. By merging this dataframe with the base dataset ($baseFREQ$) through an inner join, which contains all the policies, we can compare whether the derived clusters, represented by $cluster\_region$, align with the $Zone$ variable that was present in the dataset. It was generated a stacked histogram to visualize these comparisons.

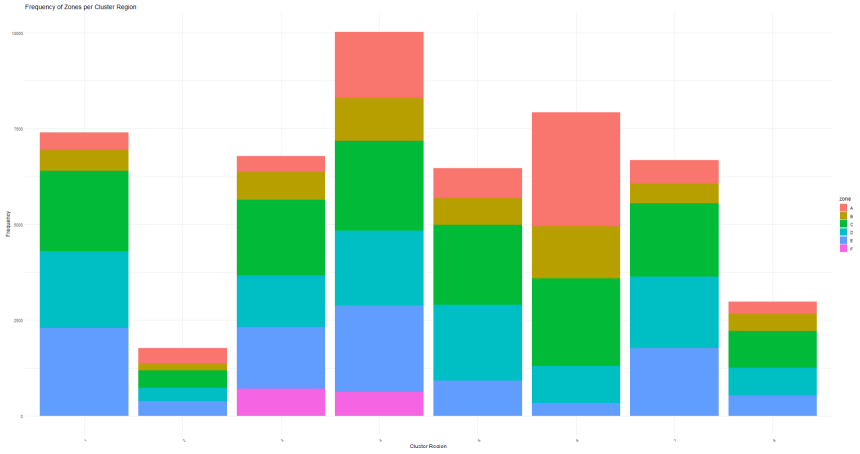


Figure 3.19: Zone vs County variable

Based on figure (3.19), we conclude that the variable $Zone$ is unrelated to the variable $cluster\_region$ due to the observed heterogeneity. The $cluster\_region$ variable was

derived from explanatory variables and indicates whether each $cluster\_region$ represents higher or lower risk for the insurer. Given this insight, we decided to use $cluster\_region$ as an explanatory variable in the models.

With this, we were able to analyse the $cluster\_region$ variable both by annual claim frequency and by average cost and exposure, where it is possible to observe great heterogeneity between the categories.

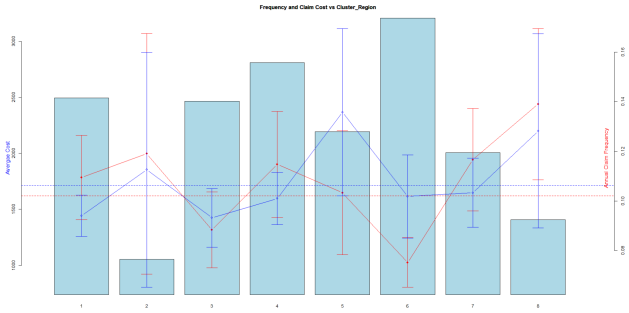


Figure 3.20: Average Claim Frequency and Claim Cost per $Region$

The analysis of average costs shows considerable heterogeneity between the different categories. Categories 5 and 8 stand out as the most expensive, while categories 1 and 3 are the least expensive. The remaining categories show values close to the average cost of the portfolio, suggesting a more homogeneous behaviour. The analysis of the annual frequency of claims confirms the heterogeneity observed. Category 8 shows the highest claims frequency, followed by categories 1, 2 and 4. In contrast, region 6 shows a significantly lower frequency of claims. By using this heterogeneity analysis, decision-makers can develop more nuanced and effective strategies, leading to improved operational efficiency performance across the portfolio. We will be using $Cluster\_Region$ as a feature to our models, but just for simplification, it will be called $Region$.

3.3 Distribution Fitting

3.3.1 Number of Claims

As mentioned before, for models where we count the number of times a discrete event occurs, we typically fit a Poisson or Negative Binomial distribution. To check whether the data follow a Poisson distribution, we need to look first whether the mean and variance are equal. In this case, the mean is 0.05394 and the variance is 0.0591472. The values are very similar, what make us test whether the data fits a Poisson distribution.

Maximum Likelihood is particularly useful for fitting Poisson and other discrete distributions because it directly maximises the likelihood of the observed data, respects the discrete nature of the data, and efficiently estimates parameters that can be used for prediction. Its robustness and flexibility make it a natural choice for analysing count data and other discrete outcomes.

For this test, we used R software with the $goodfit$ function from the vcd package using the maximum likelihood method. We obtained a p-value of 4.253155e-35, which leads us to reject the hypothesis that the data follow a Poisson distribution.

The next distribution that best fits the number of claims on a policy is the Negative Binomial distribution. Using R software, we fitted the Negative Binomial distribution to the number of claims for a goodness of fit test and obtained a p-value of 0.62228424, which lead us to not reject the null hypothesis, indicating that the number of claims on a policy fits a negative binomial distribution. Furthermore, the values estimated by the model were quite similar to the actual values, confirming that we can fit a negative binomial distribution to the number of claims.

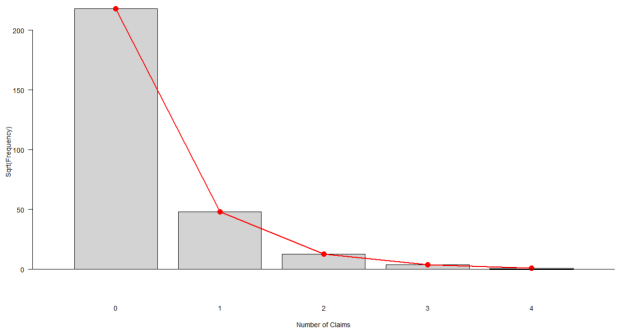
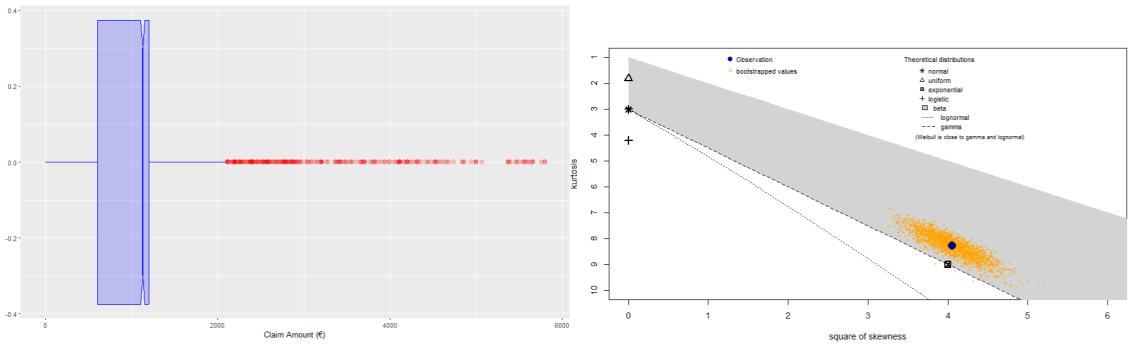


Figure 3.21: Fitting a Binomial Negative distribution

Consequently, the bar plot has the weights that were expected, since all the bars are in line with the x-axis, which means that the estimated values are similar to the actual values. Thus, the number of claims fits a Negative Binomial distribution with parameters, r=0.5246 and p=0.9073.

3.3.2 Claim Amount

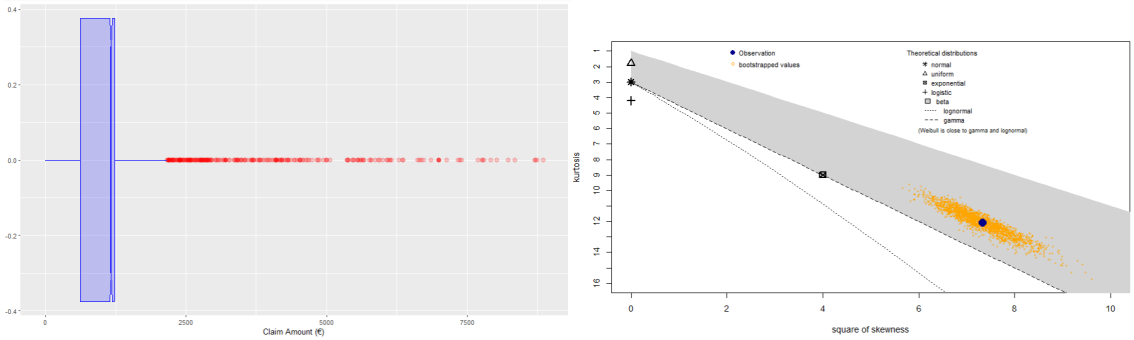
As mentioned previously, there is a large right tail representing severe claims, which have a lower probability of occurring. Therefore, in order to fit a distribution to the Claim Amount variable, we need to define a threshold L above which we will not consider claims. The threshold L has been defined based on quantiles. In order to choose the best possible threshold, we first analysed three different L values corresponding to the 96%, 97% and 98% quantiles. Each of these quantiles was analysed individually and we tried to fit a coherent distribution. For a preliminary analysis of possible distributions to fit, we examined box plots to observe the dispersion of claim amounts above the defined L_i values, and Cullen and Frey plots, which allow us to identify potential candidate distributions based on the kurtosis and skewness of the sample. We started with a threshold associated with the 96% quantile, which corresponds to a value of L_1=€5,851.2. By defining this threshold, we removed 77 claims from the initial $baseSEV$ dataframe, representing a reduction of €1,035,619 in the total claim amount, with represents an average per claim of nearly €13,500.



(a) BoxPlot (b) Cullen and Frey plot

Figure 3.22: Claim Amount with threshold L_1=96% quantile

Therefore, this dataset was defined based on the 96% quantile (L_1) shows the dispersion of the distribution, as shown in the box plot. The Cullen and Frey plot, based on kurtosis and skewness, indicates that the loss amounts are close to an exponential or gamma distribution. The boxplot and Cullen and Frey plot for the 97% quantile are very similar, suggesting a relatively stable underlying distribution within this range. By setting the cut-off point at the 97% quantile (L_2=€7,000), we eliminate 77 claims, resulting in a total claim amount of €893,709. At the 98% quantile (L_3 =€8,854.3), it is removed 39 claims which results in a total claim amount of €765,818.



(a) BoxPlot (b) Cullen and Frey plot

Figure 3.23: Claim Amount with thresold L_3=98% quantile

As we increase the threshold to the 98th percentile, we observe that the kurtosis and skewness values deviate from the Exponential distribution. This suggests that the higher the defined limit L, the lower the probability of successfully fitting a distribution. However, across all quantiles, we observe a strong tendency for the bootstrapped values to generate a line parallel to the one represented by the Gamma distribution, possibly indicating that the gamma distribution may be the most appropriate choice. Nevertheless, we will test the Exponential and Gamma distributions.

For the three datasets under consideration, defined by the thresholds mentioned above,

we tested the possibility of fitting an Exponential distribution. We used the one-sample asymptotic Kolmogorov-Smirnov test, which yielded $p\_values < 0.05$ for all tests. This led us to reject the null hypothesis, indicating that the Exponential distribution did not fit the data for any of the defined thresholds. This result suggest that the Exponential distribution does not adequately describes the data across the different thresholds. The consistent parallel trend observed with the Gamma distribution warrants further investigation of its suitability as the best-fitting distribution for our dataset.

We tested the data against the gamma distribution and obtained the results for the 3 limits already defined, as it can be showed:

Threshold	Nr Claims	shape	scale	p-value
L_1=€5,851 (quantile 96%)	1,831	1.492	815.956	0.1353
L_2=€7,000 (quantile 97%)	1,853	1.3718	932.849	0.032
L_3=€8,963 (quantile 98%)	1,869	1.269	1,053.201	1.519E-08

Table 3.6: Gamma Fitting Test Analysis

Based on our observations, we can see that only the threshold defined by the 96% quantile fails to reject the null hypothesis. However, given that the p-value associated with the L_2 threshold is very close to 0.05, which is the significance level used, we will analyse different thresholds within the interval between 96% and 97%, in increments of 0.01. This approach allows us to test these data frames 11 times and determine which threshold maximises the p-value. By performing this more granular analysis, we can gain a better understanding of the sensitivity of our results to small changes in the threshold and potentially identify a more optimal cut-off point for our statistical tests. This refined approach will give us a more comprehensive view of the behaviour of the data around the critical threshold region and help us make more informed decisions about the validity of our hypotheses.

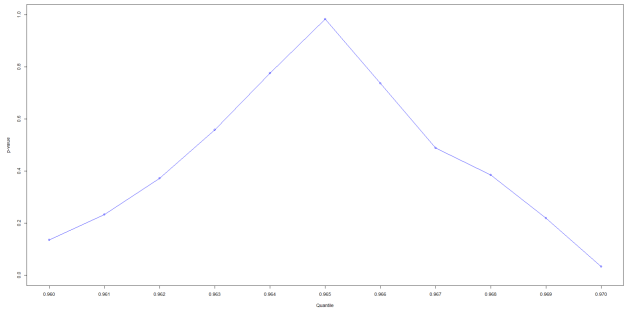


Figure 3.24: p-values maximization between 96% and 97% quantile

The plot clearly shows that the quantile that maximises the p-value between the 96% and 97% quantiles is the 96.5% quantile. Using this quantile, we obtain a data frame with a maximum value of €6,311, removing a total of 67 claims that resulted in a loss of €975,004. With this threshold, the common normal claims have an average cost of

€1,243.9, which is in contrast to the average obtained from the full dataframe. This difference highlights the impact of removing extreme values on the overall distribution of claims costs. After defining the cut-off, the fitted distribution has a shape parameter of 1.440 and a scale parameter of 863.784. To assess the goodness of fit, we can compare the theoretical values derived from this fitted distribution with the empirical values from our data. This comparison allows us to assess how well the chosen distribution models our actual claims data, particularly in the range of normal claims below the established threshold.

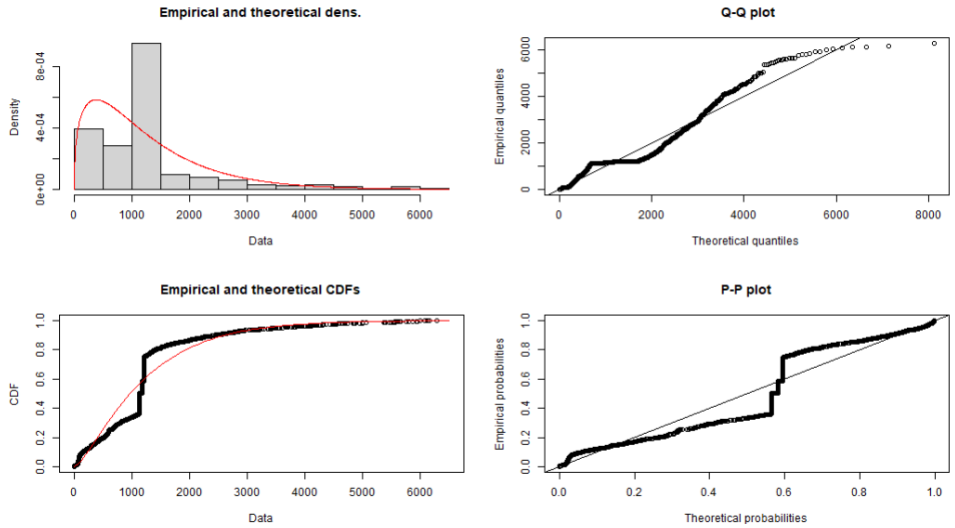


Figure 3.25: Gamma Distribution Fitting

3.4 Modelling

In this section, we will explore the methodology of the two models studied: GLM and GBM. We will look in detail at GLM modelling and present the results obtained for both frequency and severity modelling. The same process will be applied to GBM models, where we will examine the variables chosen for modelling, the differences between different hyperparameters and their added value to the model. It's worth noting that all modelling and comparison methods were implemented in Python due to its faster modelling capabilities compared to R. To construct GLM and GBM models, it's crucial to split the $baseFREQ$ and $baseSEV$ datasets into training and test sets. For this study, an 80/20 split was used for training and testing, respectively, to allow evaluation of model performance. The $train\_test\_split$ function from the $sklearn$ package is ideal for this task. By applying this function to the data sets, we obtain two new sets for each: one for training, which is used to adjust model parameters, and one for testing, which is used to evaluate the accuracy of the trained models. This random division ensures that the training and test sets are representative of the original population.

Deviance is a critical statistical measure for analysing and evaluating models. The

smaller the deviance, the better the model fits the data, i.e. it's closer to capturing the underlying relationships between variables. It is calculated using the likelihood function, which measures the probability of the observed data given the model. By comparing the deviance of different models, we can identify the one that best explains the variability in the data and therefore makes more accurate predictions. Since the claim frequency data fits a Negative Binomial distribution, we will use the deviance associated with this distribution.

$$D(y_i, \hat{\mu}) = 2 \sum_{i=1}^{n} \left(y_i \ln \left(\frac{y_i}{\hat{\mu}_i} \right) - y_i \ln \left(\frac{1 + y_i}{1 + \hat{\mu}_i} \right) \right) \tag{3.1}$$

and for claim severity modelling, it will be used the deviance of Gamma:

$$D(y_i, \hat{\mu}) = 2 \sum_{i=1}^{n} \left(-ln \left(\frac{y_i}{\hat{\mu}_i} \right) + \frac{y_i + \hat{\mu}_i}{\hat{\mu}_i} \right) \tag{3.2}$$

where y_i is each of the actual observations from the dataset under test and $\hat{\mu}_i$ is the value adjusted to that risk profile using the model.

The Generalised Error, is a metric that refines the analysis of deviance by normalising it to the number of observations in the dataset, N. This metric is calculated by dividing the total deviance of the model by the number of observations, resulting in an average deviance per observation. The use of the generalised error is particularly valuable for evaluating and comparing models, especially when working with datasets of different sizes as noted at [21].

$$GE = \frac{1}{N} D(y_i, \hat{\mu}). \tag{3.3}$$

3.4.1 GLM data treatment

As mentioned above, the $baseFREQ$ dataset containing the explanatory variables and the target variable $ClaimNb$ was divided into two sets: $baseFREQ\_train$ (39,999 policies) and $baseFREQ\_test$ (10,000 policies). The same procedure was applied to the $baseSEV$ database, resulting in $baseSEV\_train$ and $baseSEV\_test$. To ensure the reproducibility of the results, the data division was performed with a fixed $random\_state$. Before applying GLM, it was necessary to rearrange the categories of all variables. This is because GLMs use the first category of each variable as a reference for the others. We therefore ordered each variable in descending order from the $Exposure$ variable, except for the first variable, which corresponds to the category with the highest exposure in our portfolio. These categories, often referred to as the "standard insured", will serve as the basis for building the tariff structure. The categories chosen as standard are:

Feature	Standard Insured Level
Region	6
Power	6
Brand	1
Fuel	"E"
CarAge	]4, 8]
DriverAge	]32, 40]

3.4.2 GLM fitting for frequencies

For this modeling approach, it was used the glm function from the python $statsmodels$ package. The model family chosen is the Negative Binomial, as it best fits the frequency data, as previously mentioned. This adjustment was made using the $NegativeBinomial()$ function, which is also part of the $statsmodels$ package.

In this model we use the log link function to linearly link the combination of independent variables to the mean of the distribution of the response variable ($ClaimNb$). The choice of link function results in a multiplicative tariff, the one most commonly used in the insurance sector.

To refine the model, we performed Wald Test for each variable individually. This test compares the goodness of fit between two models: one with the variable in question and one without it. The variable $Power$ was the only one to be removed from the model, as the likelihood ratio test produced a non-significant p-value of $P(|z|) = 0.3858$. This suggests that, given the available data, we exclude the $Power$ variable for our model due to its low significance to the model. This simplifies the model without significantly affecting its predictive power.

We started our model with five of the six initial variables, resulting in 31 levels. Using the glm function, we estimated the β coefficients and their respective 95% confidence intervals. In addition, p-values were calculated for each level to indicate the statistical significance of the coefficients, considering a significance level of $\alpha = 0.05$. This analysis allows us to assess the influence of each variable in the model and to identify which levels are statistically significant. By examining the coefficients, confidence intervals and p-values, we can determine which variables have a significant impact on the outcome and assess the strength and direction of their relationships. This comprehensive approach provides valuable insight into the performance of the model and the relative importance of different predictors.

Variable	β	Std. Err.	z	$P(\|z\|)$	$\exp(\beta)$
Intercept	-2.5967	0.098	-26.624	≈ 0	0.0746
$CarAge$([0,2])	0.0033	0.078	0.047	0.962	1.0033
$CarAge$(]2,4])	-0.0917	0.073	-1.251	0.211	0.9124
$CarAge$(]8,12])	-0.0666	0.067	-1.001	0.317	0.9356
$CarAge$(]12,12+])	-0.0866	0.067	-1.301	0.193	0.9169
$DriverAge$(]18,24])	0.6570	0.107	6.140	0.000	1.9292
$DriverAge$([24,28])	0.0729	0.109	0.669	0.503	1.0756
$DriverAge$(]28,34])	0.0893	0.092	0.963	0.336	1.0934
$DriverAge$(]34,44])	0.0739	0.083	0.889	0.374	1.0767
$DriverAge$(]44,49])	0.0913	0.083	1.102	0.271	1.0955
$DriverAge$(]49,53])	0.0913	0.083	1.102	0.271	1.0955
$DriverAge$(]53,57])	-0.0638	0.099	-0.644	0.519	0.9382
$DriverAge$(]57,65])	-0.1198	0.093	-1.293	0.196	0.8870
$DriverAge$(]65,65+])	0.1198	0.093	1.293	0.196	1.1273
$Fuel$(D)	-0.1189	0.093	-1.263	0.206	0.8880
$Brand$(1)	0.0031	0.062	0.047	0.962	1.0031
$Brand$(2)	0.0341	0.062	0.547	0.585	1.0347
$Brand$(3)	-0.0892	0.125	-0.713	0.476	0.9147
$Brand$(4)	-0.1054	0.124	-0.849	0.396	0.8998
$Brand$(6)	-0.1624	0.126	-1.293	0.196	0.8502
$Brand$(7)	-0.2453	0.152	-1.576	0.115	0.7826
$Brand$(9)	-0.1964	0.154	-1.273	0.203	0.8218
$Brand$(10)	0.0832	0.126	0.661	0.509	1.0867
$Brand$(11)	-0.2427	0.154	-1.576	0.115	0.7845
$Brand$(12)	0.2253	0.152	1.480	0.139	1.2528
$Region$(1)	0.4545	0.081	5.607	0.000	1.5754
$Region$(2)	0.4321	0.083	5.186	0.000	1.5406
$Region$(3)	0.2593	0.091	2.849	0.004	1.2961
$Region$(4)	0.3438	0.083	4.137	0.000	1.4104
$Region$(5)	0.2681	0.091	2.936	0.003	1.3077
$Region$(7)	0.2593	0.091	2.849	0.004	1.2961
$Region$(8)	0.4740	0.107	4.437	0.000	1.6067

Table 3.7: Tariff structure with the initial frequency model, using baseFREQ_train

The process of refining the frequency model involves iteratively combining levels with the highest p-values into the standard insurer category. Initially, we identified that the level $]0, 2]$ of $CarAge$ variable had the highest p-value among all variables, making it the first to be merged with the standard insurer. We continued this iterative process, systematically combining levels with the highest p-values until all remaining levels exhibited p-values below our chosen significance level of α. By employing this methodical approach, we progressively simplified the model while retaining statistically significant predictors. The process allowed us to strike a balance between model complexity and explanatory power. Ultimately, we finished at a refined model comprising 13 frequency levels, each demonstrating statistical significance compared with "Standard Insured" categories. This streamlined model not only enhances interpretability but also focuses on

the most influential factors affecting our outcome of interest, providing a more robust and efficient framework for analysis and prediction.

Variable	β	Std. Err.	z	$P > \|z\|$	exp(β)
Intercept	-2.5517	0.068	-37.751	≈ 0	0.0779
$CarAge$(]12,12+])	-0.2017	0.060	-3.345	0.001	0.8173
$DriverAge$([18,24])	0.5886	0.097	6.085	0.000	1.8014
$DriverAge$(]57,65])	-0.1676	0.082	-2.049	0.040	0.8457
$DriverAge$(]65,65+])	-0.1742	0.079	-2.197	0.028	0.8401
$Fuel$(D)	0.9964	0.046	2.107	0.035	1.1012
$Brand$(12)	-0.2427	0.063	-3.844	0.000	0.7845
$Region$(1)	0.4572	0.081	5.662	0.000	1.5796
$Region$(2)	0.4590	0.139	3.307	0.001	1.5825
$Region$(3)	0.2265	0.085	2.666	0.008	1.2542
$Region$(4)	0.3633	0.079	4.608	0.000	1.4381
$Region$(5)	0.2682	0.088	3.047	0.002	1.3076
$Region$(7)	0.4372	0.088	4.983	0.000	1.5484
$Region$(8)	0.4915	0.103	4.782	0.000	1.6348

Table 3.8: Tariff structure with the final frequency model, using baseFREQ_train

The $\boldsymbol{\beta}$ estimates for each class represent the relative risk of claims for an individual compared to a standard insured. Negative values indicate that contracts with a particular characteristic tend to have a lower risk of reporting a claim, while positive values indicate a higher likelihood of claims.

The expected claim frequency for an insured is calculated by multiplying the base frequency (of the standard insured) by $exp(\beta_i)$, where β_i corresponds to the level of the feature in question. In the model, for example, the base frequency is 0.0779 claims per year. For an individual aged between 18 and 24 and the other features equal to the "standard insurer", the model estimates an annual claim frequency of 0.1403, as the $exp(\beta_{(DriverAgeB6)[18,24]}) = 1.8014$ is multiplied by the base frequency.

This approach allows us to analyse the individual impact of each characteristic on the number of claims. Therefore, we can obtain a precise estimate of the frequency of claims for an individual with a particular profile. This allows us to see which profile has the highest risk:

$$\text{CarAge} \in [12, 12+[, DriverAge \in [18, 24],$$
$$\text{Fuel = "D", Brand=1, Region=8.}$$

Similarly, we can see the profile that represents the lowest risk, which has the following characteristics:

$$\text{CarAge} \in]4, 8], DriverAge \in [65, 65+[,$$
$$\text{Fuel = "E", Brand=12, Region=6.}$$

As previously mentioned, the model was built based on a training dataset. In order to test the model and to compare the two models later, we applied the adjusted model to the baseFREQ_test dataset using Python's $predict()$ function.

3.4.3 GLM fitting for Claim Amount

The process of fitting the predictive model for the Claim Amount variable follows a similar approach to that used for frequency modelling. We first test for non-significant variables in the model using the log likelihood test. We then analyse each level of the significant variables to test their individual significance in order to obtain a model that is well fitted to the data. Once we have a fitted model, the next step is to use the test data set. Using the $predict()$ function, we fit the final model to the test data and evaluate its performance by examining the deviance and overall error. As before, the model is fitted using the glm function from the $statsmodels$ package, which allows us to perform Gamma regression fitting. With this method in place, we start by examining the model that includes all the potential predictor variables. However, once the model was complete, there were many variables that were not significant to the model at all levels, which led us to perform several Wald tests in order to include only the significant variables in the model.

Test	p_value
Model 1 vs Model 2	0.4568
Model 2 vs Model 3	0.7217
Model 3 vs Model 4	0.6712
Model 4 vs Model 5	0.5482

Table 3.9: p_values of Wald test

where,

Model 1 - Model with all variables;
Model 2 - Model without Fuel;
Model 3 - Model without Fuel and Power;
Model 4 - Model without Fuel, Power and CarAge;
Model 5 - Model without Fuel, Power, CarAge and Region.

It can be seen that the variables $Fuel$, $Power$, $CarAge$ and $Region$ are not significant, leaving only two explanatory variables in the model.

Variable	β	Std. Err.	z	$P > \|z\|$	exp(β)
Intercept	6.9218	0.061	113.137	≈ 0	1,015.847
$DriverAge$([18,24])	0.1849	0.095	1.947	0.052	1.203
$DriverAge$(]24,28])	0.0861	0.088	0.981	0.327	1.090
$DriverAge$(]28,32])	0.0659	0.088	0.752	0.452	1.068
$DriverAge$(]40,44])	0.0421	0.071	0.594	0.552	1.043
$DriverAge$(]44,49])	0.2431	0.077	3.164	0.002	1.275
$DriverAge$(]49,53])	0.1065	0.093	1.143	0.253	1.112
$DriverAge$(]53,57])	0.0090	0.083	0.108	0.914	1.009
$DriverAge$(]57,65])	0.2312	0.086	2.701	0.007	1.260
$DriverAge$(]65,65+])	0.1028	0.088	1.169	0.242	1.108
$Brand$(2)	-0.0104	0.084	-0.123	0.902	0.990
$Brand$(3)	-0.2245	0.103	-2.176	0.030	0.799
$Brand$(4)	-0.0458	0.092	-0.593	0.553	0.955
$Brand$(5)	0.3474	0.129	2.701	0.007	1.415
$Brand$(6)	0.1789	0.091	1.972	0.049	1.196
$Brand$(7	-0.0483	0.108	-0.448	0.654	0.953
$Brand$(9)	0.2465	0.122	2.017	0.044	1.279
$Brand$(11)	-0.1460	0.143	-1.020	0.308	0.864
$Brand$(12)	-0.1645	0.143	-1.148	0.251	0.848
$Brand$(13)	-0.1160	0.143	-0.814	0.416	0.891
$Brand$(14)	0.7600	0.411	1.851	0.064	2.138

Table 3.10: Tariff structure with the initial severity model, using baseSEV_train

Using the same iterative method as in the frequency modeling, we removed the levels with the higher p_values. In this way, we obtained a model with 8 levels where for all but one of the levels we have a significance lower than $\alpha = 0.05$.

Variable	β	Std. Err.	z	$P > \|z\|$	exp(β)
Intercept	6.9563	0.033	213.132	≈ 0	1050.724
$DriverAge$(]18,24])	0.1458	0.084	1.734	0.083	1.157
$DriverAge$(]49,53])	0.1905	0.067	2.836	0.005	1.210
$DriverAge$(]65,65+])	0.1905	0.074	2.585	0.010	1.210
$Brand$(2)	0.1393	0.052	2.700	0.007	1.150
$Brand$(4)	0.2426	0.107	2.258	0.024	1.275
$Brand$(10)	0.3501	0.125	2.807	0.005	1.419
$Brand$(12)	0.2517	0.060	4.192	0.000	1.286
$Brand$(14)	0.7665	0.408	1.881	0.060	2.152

Table 3.11: Tariff structure with the final severity model, using baseSEV_train

The GLM model fitted for $ClaimAmount$ shows fewer significant variables and levels compared to the model fitted for claim frequency. This difference is mainly due to the heterogeneous nature of motor claims, where severity is influenced by fewer factors than frequency, specially in TPL portfolio. The retention of the age category for drivers between 18 and 24 years of age increases the model's deviance. This retention is justified by the preliminary variable analysis which showed that this age group had a significantly higher

average claim cost compared to the standard insured.The decision to retain this age group, despite its impact on variance, highlights the importance of balancing statistical fit with practical insights. In addition, the inherently higher variability of claim size often requires more parsimonious models to effectively capture the key drivers. It's worth noting that the performance of this model on the test portfolio is discussed in the model testing chapter, providing further insight into its predictive capabilities and robustness across different datasets.

3.4.4 GBM Target Enconding

Target encoding is a technique used to transform categorical features into numerical representations that are suitable for machine learning algorithms. In the context of GBM, target encoding can be particularly effective because it directly captures the relationship between the categorical variable and the target variable. To further improve the performance of the model, we will numerically rank the categorical variables based on their exposure in the dataset. By assigning numerical values to categories according to their exposure, we introduce a hierarchical structure that can be exploited by the Machine Learning algorithm. This approach can be particularly beneficial when dealing with unbalanced datasets or when certain categories have a more significant impact on the target variable.

With this approach, we will continue to use the numerical variables, i.e. $CarAge$ and $DriverAge$. On the other hand, the variables $Power$, $Brand$ and $Region$, although categorical, are represented by numerical categories and will therefore remain in their current form. However, the variable $Fuel$ is transformed into $FuelT$, where "T" stands for target. In this transformation, the category "E" is represented by 0 and the category "D" is represented by 1.

3.4.5 Claim Frequency with GBM Modelling

We will apply the GBM model to model the claim frequency in the portfolio, using the same $baseFREQ\_train$ dataset as previously used in the GLM application for comparison. For the purposes of performance tuning and cross-validation, we will include all the variables in our model, but the variables included in the definition of the final model will be examined later. The target of the model is the claim frequency. As the GBM does not have a native offset, it was decided to adjust the target and model weights accordingly. This approach helps to correct for the lack of offset by ensuring that the model gives greater weight to observations with greater exposure, which is essential to capture the variability in claims frequency without bias. Thus, the decision to use claims frequency as the target and exposure as the weight ensures that the model is correctly calibrated even without the use of an explicit offset.

- Response variable: $ClaimNb/Exposure$

- Predictor variables: $CarAge$,$DriverAge$, $Brand$, $Region$, $FuelT$ and $Power$
- Weights: $Exposure$

Before starting the GBM training process, the first crucial step is to choose the appropriate hyperparameters. These hyperparameters include the number of trees, the maximum depth of the trees (max_depth), and the learning rate (learning_rate). Careful selection of these hyperparameters can have a significant impact on the performance of the model, and a technique called grid search is often used for this purpose.

Grid search is a systematic approach to testing different combinations of hyperparameters to identify the configuration that best fits the specific data set. In the context of GBM, three main hyperparameters are studied:

- $Number\ Of\ Trees \in \{100, 200, 500\}$: Represents the number of individual decision trees that make up the final model. A larger number of trees generally results in a more complex model with greater learning capacity.
- $Max\_depth \in \{1, 2, 3, 4, 5\}$: Defines the maximum depth of each decision tree. A greater depth allows the model to capture more complex relationships between variables.
- $Learning\_rate \in \{0.005, 0.01, 0.02, 0.05\}$: It controls how well each successive tree corrects the errors of the previous trees.

Having defined the grid of possible values for these hyperparameters, the next step is to evaluate how each combination performs in terms of model performance. It is common to use a cross-validation technique to do this. In this particular case, K-fold cross validation was used, where the dataset is divided into $K = 5$ subsets. For each iteration, one of the K folds is reserved as a validation set, while the remaining $K - 1$ folds are used to train the model. For each fold, the GBM model is fitted with the specific hyperparameters being tested. During this fitting process, the model predictions are compared to the actual values and the Negative Binomial Deviance is calculated. With the Negative Binomial Deviance in hand, it's possible to calculate the Generalised Error for each fold.

After performing the process for all K folds, the average Generalised Error is calculated to provide a robust evaluation of the model's performance for that specific combination of hyperparameters. This cycle of training, prediction and evaluation is repeated for all possible combinations of hyperparameters defined in the grid search. At the end of the process, there is a clear understanding of which combinations of Number of Trees, Max Depth and Learning Rate produce the best model, balancing the ability to capture complex patterns in the data with the need to avoid overfitting, culminating in the selection of the best hyperparameters for the GBM. This comprehensive approach ensures that the final model is not only well tuned, but also robust and generalise to new data.

These are the results obtained for the 5 best parameter combinations:

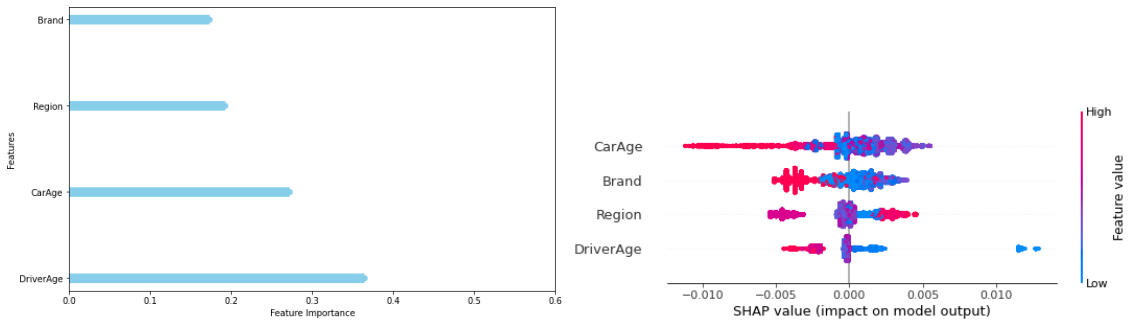
Learning Rate	Number Of Trees	Max Depth	Negative Binomial Generalised Error
0.05	**100**	**4**	**0.327297**
0.02	200	4	0.327301
0.01	500	3	0.327336
0.005	500	5	0.327342
0.05	100	3	0.327359

Table 3.12: Optimal tunning parameters

Thus, the optimal parameters for this model are the ones that minimizes the Negative Binomial Generalised Error. After the detailed process of adjusting the parameters of the GBM model using techniques such as grid search and cross-validation, we can move on to an equally important stage: analysing the model. This stage allows us not only to understand how the model arrived at its predictions, but also to identify which features have the greatest influence on the final result.

Two powerful tools stand out here: feature importance and SHapley Additive exPlanations (SHAP) values. Feature importance gives us an overview of the relevance of each feature to the model as a whole, i.e. which features contributed most to error reduction during training.

However, feature importance does not tell us whether each feature tends to increase or decrease prediction, and this is where SHAP analysis comes in. SHAP scores provide a more detailed and individualised explanation of the predictions made by complex models. The SHAP score assigns each variable a specific contribution to the prediction, taking into account all possible interactions between variables. This makes it easier to understand how each variable influenced a particular prediction, and provides a detailed and precise analysis of the model's behaviour in relation to the variables analysed. Together, these two approaches provide a robust and comprehensible analysis of the factors that most influence predictive modelling decisions.



(a) Feature importance

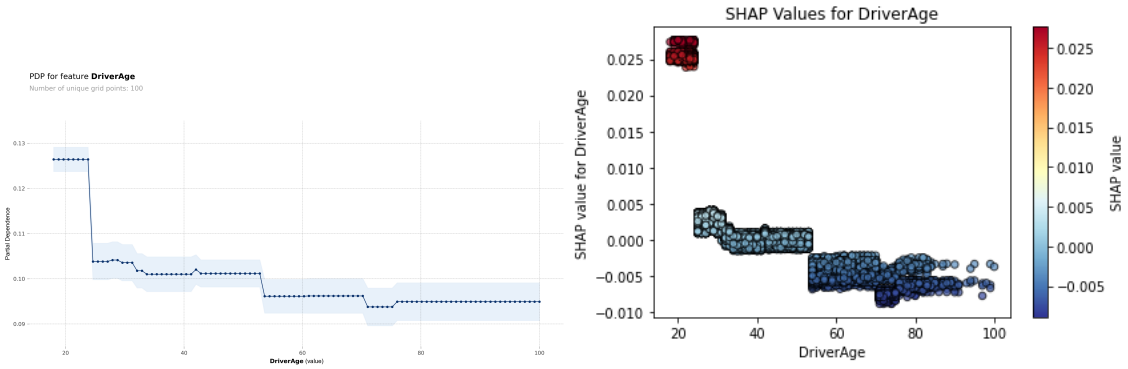
(b) SHAP Values

Figure 3.26: Model prediction analysis

The features used in the model with optimal parameters are not the same as those selected in the GLM for frequency. This is due to the fact that the feature importance value is analysed in a normalised way, i.e. by dividing the feature importance of each feature by the total. In this way, we remove the features that have a value below 5%, so we are left with a model with 4 features, removing the $Brand$ and $FuelT$ features. As expected, $DriverAge$ is by far the most important variable in our model, which confirms the work developed in the GLM. The $Region$ variable also has some importance, but not comparable to the $DriverAge$ variable. On the other hand, it can be seen that the effect of lower age on the model output is positive, indicating that lower age in $DriverAge$ leads to higher claim frequency, which was expected based on previous analysis. The SHAP analysis allows a better examination of numerical variables such as $DriverAge$ and $CarAge$, where it is possible to observe the discrepancies between high and low values of the variables and their impact on the model output. To analyse all the features in our model, we will use PDPs along with the SHAP analysis for each feature.

PDP is a powerful tool for interpreting Machine Learning models, especially in scenarios where the complexity of the model makes it difficult to directly understand its decisions. Essentially, the PDP visualises the marginal effect of one or more predictor variables on the outcome of the model, while holding all other variables constant. This allows us to observe how changes in a particular variable affect the model's prediction, providing an intuitive view of the relationship between features and the target variable. By combining PDP with SHAP analysis, we can gain a comprehensive understanding of how each feature contributes to the model's predictions and how they interact, providing valuable insights into the underlying patterns in our claims frequency data.

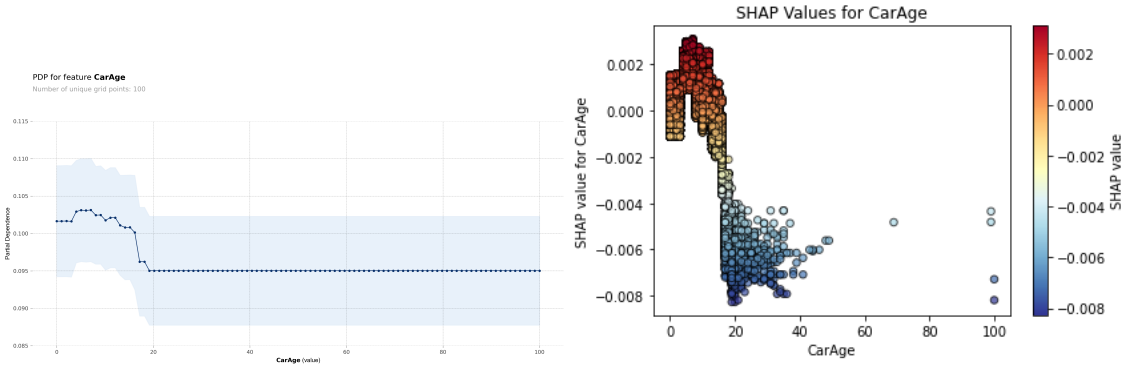
Thus, starting with the $DriverAge$ feature, both the PDP and SHAP value analyses show a positive effect on model output for younger ages, indicating that younger drivers are more likely to have claims. There is a negative trend in the frequency of claims, indicating that as the age of the driver increases, the frequency of claims decreases. For middle-aged drivers (approximately 25 to 70 years old), SHAP values are close to zero, indicating that age in this range does not significantly affect claim frequency in the model. Drivers between the ages of 25 and 70 do not have a significant impact on the model, as indicated by the SHAP value.



(a) PDP (b) SHAP graph

Figure 3.27: *DriverAge* Analysis

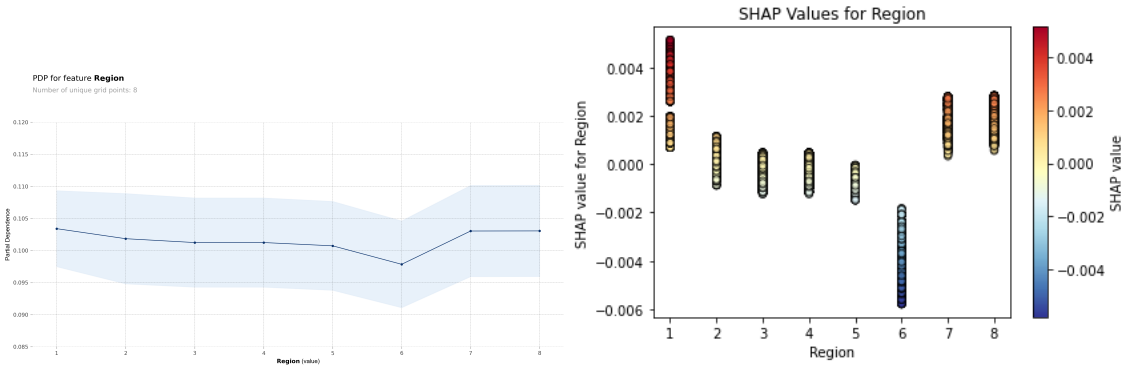
For the *CarAge* variable, the same trend can be observed in both PDP and SHAP analyses, indicating that cars up to 20 years old tend to have a higher loss frequency than cars over that age. This is also largely reflected by the exposure of cars older than 20 years in our portfolio. It can be observed that new cars in our portfolio have a low claim frequency. However, this frequency tends to increase after a few years and then decrease again.



(a) PDP (b) SHAP graph

Figure 3.28: *CarAge* Analysis

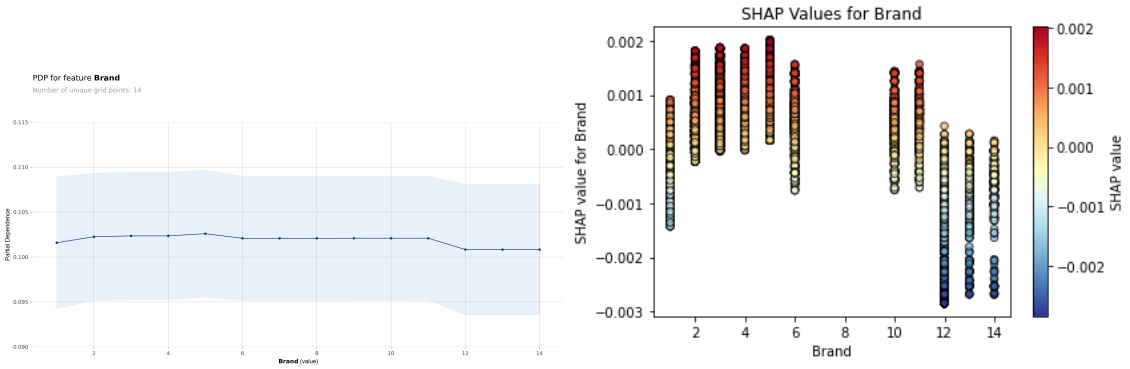
The *Region* variable was expected to show a greater difference in the PDPS, as it was generated by cluster analysis. With regard to the SHAP analysis, it's possible to observe that levels 1, 7 and 8 have a positive impact on the model output, meaning a higher frequency of claims. On the other hand, region 6 has a negative impact on the model output, indicating a decrease in claim frequency for policies in this Region. Levels 2, 3, 4 and 5 have little effect on the model output.



(a) PDP (b) SHAP graph

Figure 3.29: *Region* Analysis

Before analysing the *Brand* variable, it should be noted that this variable is a categorical variable and does not have categories from 7 to 10, which is why we see a gap in the SHAP analysis figure, a trend that is not present in the PDP. On the other hand, the SHAP analysis does corroborate the PDP as it shows how little significance this variable has in our model as the SHAP values of all the variables are very close to zero. However, the category 12 is the one that has the greatest impact on the output model, where we can see that this variable brings a lower claim frequency to the analysis.



(a) PDP (b) SHAP Graph

Figure 3.30: *Brand* Analysis

3.4.6 Claim Severity with GBM Modelling

To model claim severity using the GBM model, we will adopt an approach similar to the one used for frequency modelling. We will begin by constructing a GBM model incorporating all the features. The selection of optimal parameters will be conducted through Grid Search and cross-validation. Once the model is defined, we will analyze the importance of variables to understand which factors most significantly influence severity. Additionally, we will use SHAP values to obtain a more detailed interpretation of the model results, enabling us to identify how each variable contributes to the prediction of individual claim severity. For this model, we will not use the weights used in the frequency model. Consequently, we will focus on the predictor variables and the response variable without considering weights.

- Response variable: $ClaimAmount$
- Predictor variables: $CarAge$, $DriverAge$, $Brand$, $Region$, $FuelT$ and $Power$

With this, the Grid Search was carried out with the following parameters in order to obtain the optimal parameters

- $Number\ Of\ Trees \in \{100, 200, 500\}$
- $max\_depth \in \{1, 2, 3, 4, 5\}$:
- $learning\_rate \in \{0.005, 0.01, 0.02, 0.05\}$

It should be noted that this search for the optimum parameters that minimise the Deviance and consequently the Generalised Error, was based on the same process used in the claim frequency through cross validation. The only difference was the use of Gamma Deviance instead of Negative Binomial Deviance due to the distribution adjusted. Thus, the 5 best combinations that minimise the Generalised Error are:

Learning Rate	Number Of Trees	Max Depth	Gamma Generalised Error
0.02	**200**	**1**	**0.816676**
0.005	100	1	0.81668
0.01	500	1	0.816684
0.005	100	3	0.816813
0.005	500	1	0.81691

Table 3.13: Optimal tunning parameters

So, given that the search for the optimum parameters takes into account all the features, it is beneficial to see the importance of the significant features in the GBM model. As with the claim frequency model, we obtained the normalised feature value for each of the features in our model and chose the features with values greater than 5% as a proportion

of the total feature selection value. In this way, we chose 3 features for our model: $Brand$, $DriverAge$ and $CarAge$.

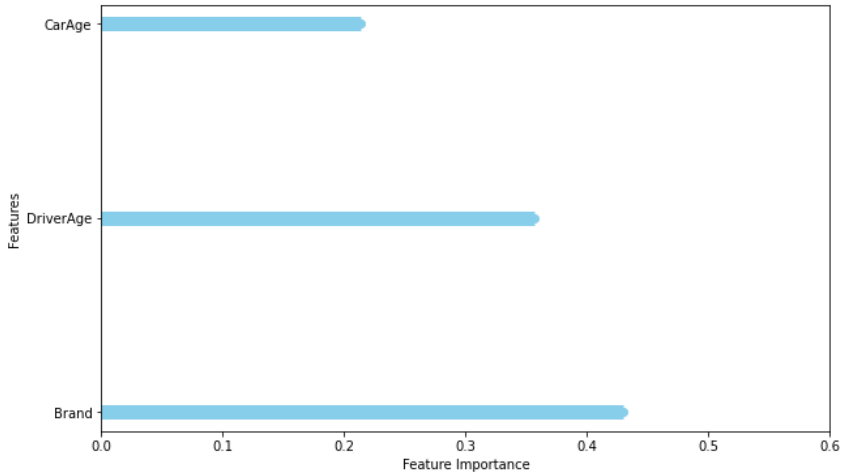


Figure 3.31: Feature Importance plot for chosen features

That said, we were able to analyse the PDP to get a better idea of how each level of the two variables influences the model output.

We begin with the numerical variable $DriverAge$, where it is possible to observe the behaviour of the $DriverAge$ variable as expected given that younger customers and this severity tends to decrease up to borrowers aged 30. After that, the severity tends to decrease until the age of 45. After that, the severity tends to increase again between the ages of 45 and 60. From 65 to 80 there is another impact on the model.

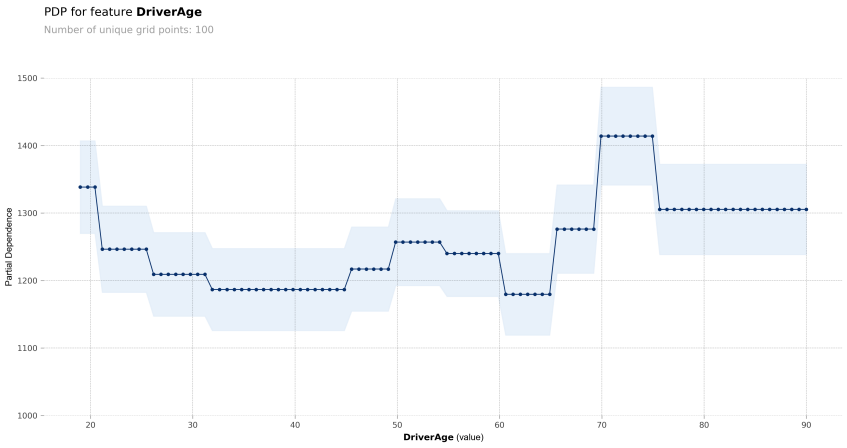


Figure 3.32: PDP for feature $DriverAge$

On the other hand, it can be seen that the behaviour of the $Brand$ variable is very similar to the real value, since classes 10 and 11 are the most severe levels, and this behaviour is reflected in the following figure:

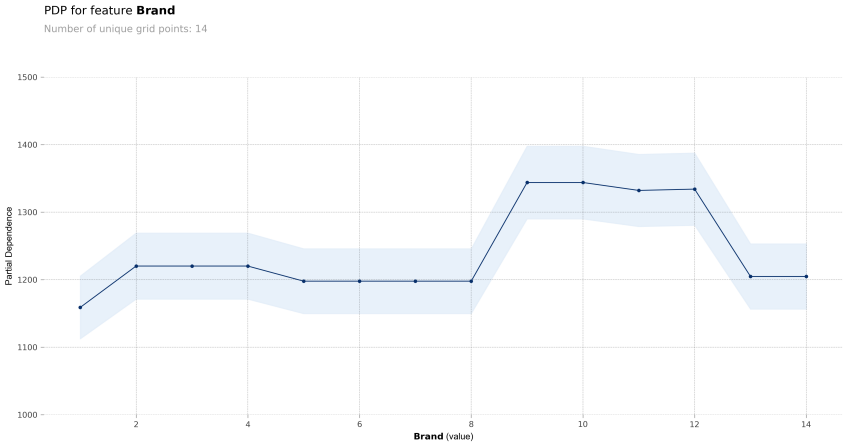


Figure 3.33: PDP for feature $Brand$

Ultimately, the $CarAge$ feature shows a greater impact on the model's output between the ages of 2 and 10. After the age of 20, there is no major impact on the model's output, which is to be expected given the few observations in our portfolio.

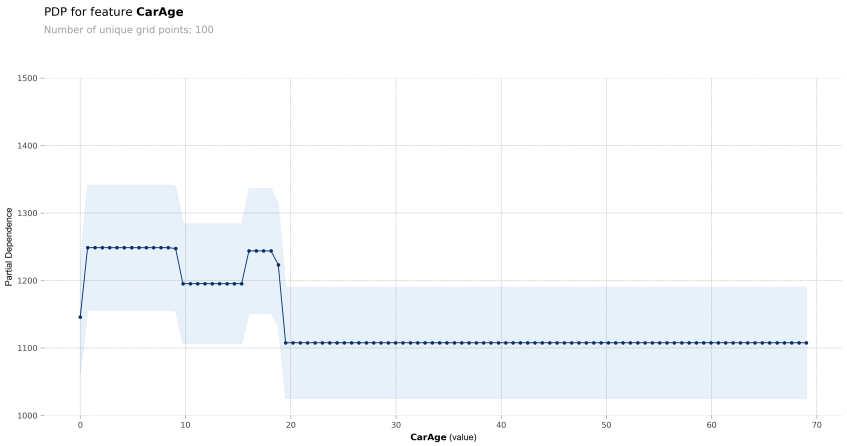


Figure 3.34: PDP for feature $CarAge$

With the model estimated and trained, we will test it, and this test will be evaluated in the next section.

3.5 Modelling Results

3.5.1 Deviance and Generalised Error Analysis

For all the models, the Deviance was calculated based on the training dataset and the test dataset. Based on the Deviance, we calculated the Generalised Error. As already mentioned for the GLM and GBM models that we fitted to the claims frequency, we used the Deviance of the Negative Binomial due to the fit of this distribution to our data. On the other hand, as mentioned above, we used the Gamma Deviance for the GLM and GBM models that attempt to model Severity.

That said, it is possible to compare the Deviance of the 4 models fitted, both on the basis of the training dataset and the test dataset:

Model	Frequency		Severity	
	Train Set	Test Set	Train Set	Test Set
GLM	14331.40	3675.05	1190.98	282.85
GBM	14308.30	3668.39	1170.11	276.65

Table 3.14: Total Deviance

The table above shows that, based on the training dataset (80% of the total data from the $baseSEV$ and $baseFREQ$ datasets), the GBM had a consistently lower Deviance compared to the other models evaluated.

This reduction in Deviance suggests that the GBM is more effectively capturing the explanatory variables and their interactions, leading to a better prediction of the target values in both datasets. Consequently, we can infer that the GBM is potentially the most suitable model for the data in question, providing more accurate predictions and thus being preferable for future applications, such as predicting event frequencies ($baseFREQ$) or the severity of claims ($baseSEV$).

This can be seen in the table where we indicate the Generalised Error:

Model	Frequency		Severity	
	Train Set	Test Set	Train Set	Test Set
GLM	0.35829	0.3675	0.80799	0.77073
GBM	0.35772	0.36684	0.79383	0.75381

Table 3.15: Generalised Error

When comparing frequency or number of claims models, the GBM has a lower Generalised Error, measured on an unseen portfolio of policies, than the GLM. Similarly, for severity or claim amount models, the GBM models outperform both of their GLM counterparts.

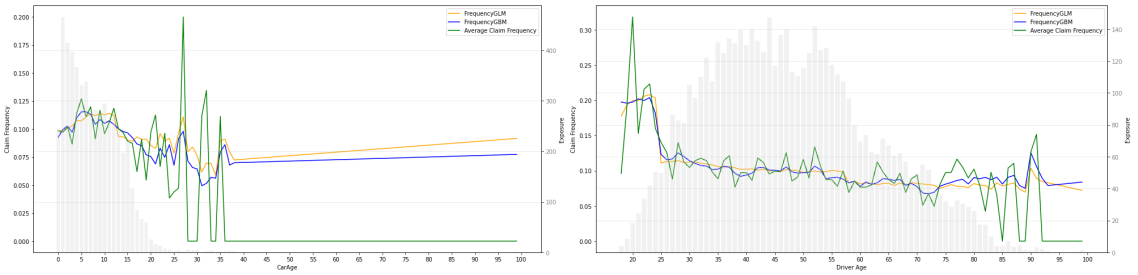
3.5.2 Statistical Comparison on Frequency and Severity

For analysis purposes, we will only consider the predictions made with the training dataset. We will compare the fitted and tested models in terms of both frequency and severity. First, by evaluating the models fitted for frequency, we were able to obtain the following table:

Empirical Claim Frequency	GLM Claim Frequency	GBM Claim Frequency
0.104742	0.103498	0.102347

Table 3.16: Frequency Analysis

We can see that the claim frequency estimated by the GBM is the lowest compared to the GLM and the claim frequency observed in the portfolio.



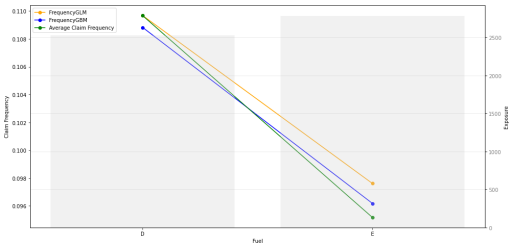
(a) CarAge feature (b) DriverAge feaure

Figure 3.35: Average claim frequency comparison in impirical vs glm vs gbm

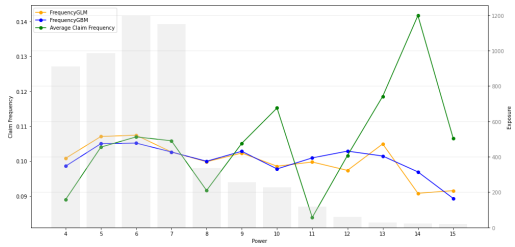
We can then analyse and compare the behaviour of the two models separately by feature, starting with the numerical variables, $DriverAge$ and $CarAge$. When analysing at the GLM claim frequency predictions, it is very clear the categorical divisions made before modelling GLM. This can be observed, on $DriverAge$ feature, where the ages between 18 and 24 have a very high frequency, as would be expected, and the gap between 24 and 25 is very abrupt and does not represent reality, since at the age of 25 the claim frequency is still high. The GBM replicates the behaviour of the variables better than the GLM, which is always very centralised by the divisions made. The blue line representing the GBM predictions seems to track the actual average claim frequency (green line) more closely across different age groups. This suggests that the GBM is better at capturing the complex, non-linear relationships between driver age and claim frequency. The orange line, representing the GLM predictions, tends to smooth out the variations in the claim frequency.

For drivers in their early 20s, the empirical claim frequency (green line) shows significant spikes that are better captured by the GBM model. The GLM underestimates these spikes, indicating a lack of flexibility in modelling these fluctuations. For middle-aged drivers (around 30-60), both models perform relatively well, as the claim frequency is

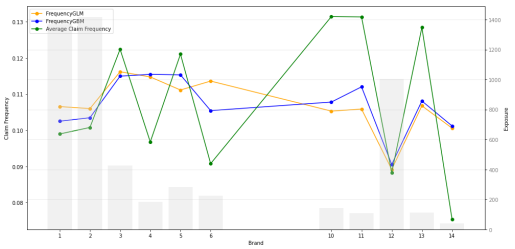
more stable and closer to linear. In the older age groups (70+), the GBM model continues to fit the actual data better, capturing the small increases and decreases, while the GLM remains more linear and does not fit these patterns well. The plot clearly shows that the GBM model, which can capture non-linear relationships and interactions, provides a better fit to the actual claim frequency data than the GLM. The GLM, with its inherent linearity, does not fit well to the more complex, non-linear patterns observed in claim frequency across age groups. This highlights the limitations of linear models such as the GLM in scenarios where the relationship between variables is not strictly linear.



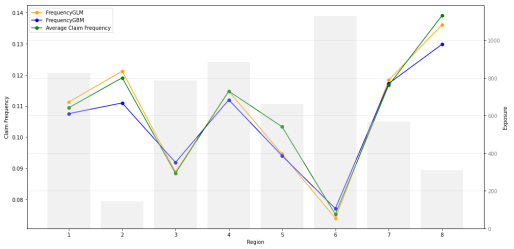
(a) Fuel feature



(b) Power feature



(c) Brand feature



(d) Region feature

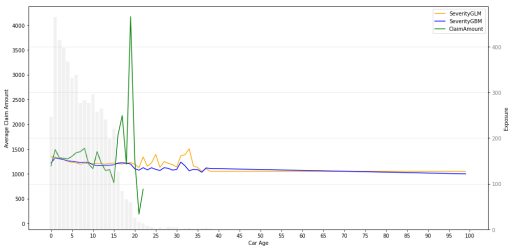
Figure 3.36: Average claim frequency comparison in impirical vs glm vs gbm

With regard to the categorical variables, the *Power* variable is not significant in the GLM and GBM models and is reflected in the behaviour of the claim frequency of the two models when compared to the estimated claim frequency. The *Region* data fits the estimated frequency quite well, unlike *Brand*, which has only one of the levels that was significantly different from the Standard Insured defined in the GLM and the feature importance was quite high in the GBM.

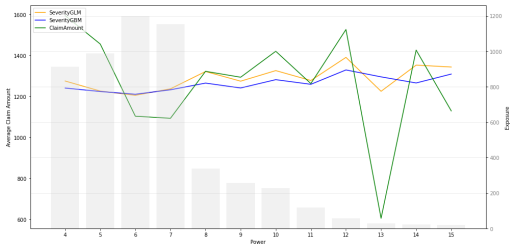
On the other hand, we must analyse the predictions for severity, as it can be shown as follow:

Average Claim Amount	Claim Amount GLM Prediction	Claim Amount GBM Prediction
473,715	457,162	453,149

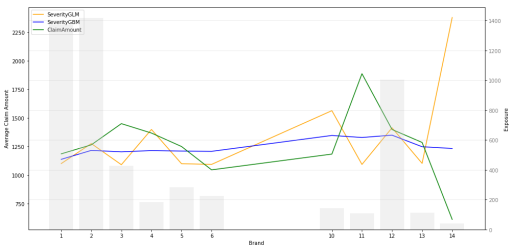
Table 3.17: Total CLaim Amount Prediction



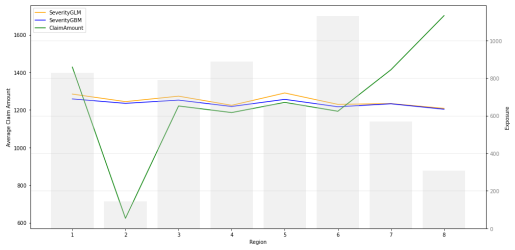
(a) CarAge feature



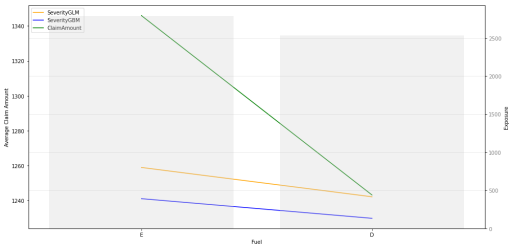
(b) Power feature



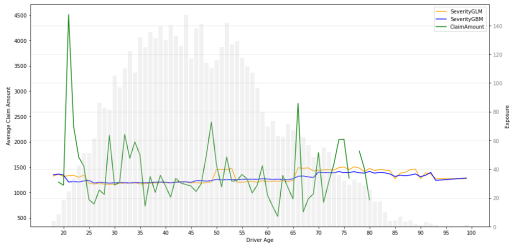
(c) Brand feature



(d) Region feature



(e) Fuel feature



(f) DriverAge feature

Figure 3.37: Average Claim Amount vs GLM vs GBM

In the comparative analysis of the severity models, we observed an interesting phenomenon related to the significant variables $DriverAge$ and $Brand$. Remarkably, these variables show a better fit to the dependent variable $ClaimAmount$ in the GLM model than in the GBM model. However, it is always important to note that the GBM showed better results when interpreting the $ClaimAmount$ variable, which can be seen in the tables 3.14 and 3.15. Although this result is interesting, it should be interpreted with caution. The apparent superiority of the fit of the current model is mainly due to the limited size of the test data set used in this analysis.

3.5.3 Pure Premium Analysis

Given the predictions for claim frequency and severity, we can obtain the pure premium for each policy and compare the results for each model.

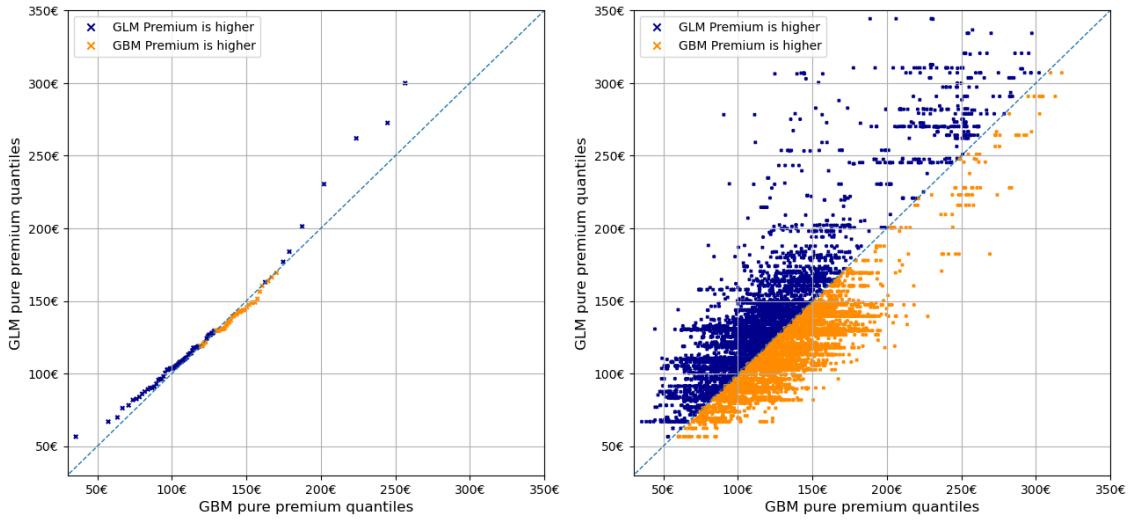
$$Pure\ Premium = \mathbf{E}(N)\mathbf{E}(Y)$$

Let $\mathbf{E}(N)$ be the expected number of claims and $\mathbf{E}(Y)$ be the expected cost per claim. That said, we were able to understand the behaviour of the premiums adjusted by the GLM and GBM using this statistical analysis:

	GLM Pure Premium	GBM Pure Premium
Portfolio	457,160	453,150
Mean	128.59	126.09
Standard Deviation	41.30	37.37
25% quartile	104.52	101.79
75% quartile	140.79	143.62
Minimum	56.56	35.57
Maximum	527.84	317.40

Table 3.18: Pure Premium Analysis

In an attempt to compare the premiums generated by the GLM and GBM models, a quantile analysis was carried out using a scatter plot. This visualisation shows the relationship between the premiums calculated by both methods, where the proximity of the points to the diagonal line indicates greater similarity between the models. The closer the points are to this line, the more aligned the premium estimates of the two models are.



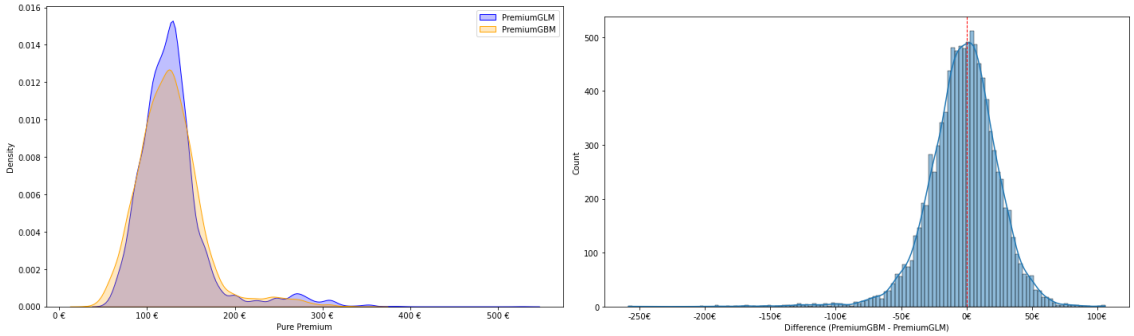
(a) Quantiles comparision

(b) All the policies from test set

Figure 3.38: Pure Premiums Scatter plot

When analysing the results, we can notice that the GLM tends to penalise policies with higher premiums in comparison with the GBM, which has a more balanced approach. For lower premiums, the GLM is also more rigorous, which may result in a more conservative risk assessment. This difference in approach is crucial to understanding how each model deals with policy pricing. It is also important to note that there are specific ranges in which the GBM shows higher premiums than those adjusted by the GLM. In the 45 to 49 and

59 to 92 quantiles, the premiums calculated by the GBM are higher, while in all the other quantiles, the GLM maintains higher premiums. This analysis reveals important nuances in pricing between the two models, indicating that each can capture different aspects of the risk associated with policies, which can influence strategic decisions in portfolio management.



(a) Density plot of two premiums models

(b) Difference plot between models

Figure 3.39: Pure Premium comparison

The predominant concentration of premiums around €120 is evident for both models, indicating a convergence in the assessment of average risks. However, the GLM exhibits a slightly higher density in this central range, which can be attributed to the linear nature of the GLM, which tends to produce more stable and predictable results. On the other hand, the GBM demonstrates greater flexibility in modelling, which can be seen in the tails of the distribution. This ability to capture more extreme variations results in a slightly more dispersed distribution, with occasionally higher premiums for certain risk profiles. This feature reflects the GBM ability to identify and quantify complex, non-linear patterns in the data, potentially offering more granular and adaptive pricing. These subtle but crucial differences between the two models have significant implications for a pricing strategy.

4

Conclusions

This study compared the performance of the GBM and GLM models in modelling the frequency and severity of claims in a TPL portfolio, contributing to the advance of research in the non-life insurance market.

The results obtained clearly show that the GBM outperformed the GLM in both the frequency and severity of claims models. This superiority was evidenced by consistently lower deviance values in the training and test datasets for the GBM. These results are consistent with previous studies, such as the work of Victor Martinez in 2021 [22]. In the study of an automobile insurance portfolio, the author also concluded that GBM models consistently outperformed GLMs, with lower values of deviance across train and test datasets. These findings suggest that we can rely on ensemble tree models, such as the GBM, to carry out robust and accurate tariff studies. While these results consistently demonstrate the superiority of the GBM model over the GLM across all datasets, it is important to consider the nuanced insights provided by Carina Clemente paper [23]. In the authors paper, it was found that the relative performance of GBM and GLM models can vary across different datasets. While GBM often outperforms GLM in terms of deviance, there are cases where the superiority of GBM is less pronounced, specially in claim severity modeling. In summary, ongoing paper and comparative analyses between ensemble tree models and GLM models remain crucial. Such investigations will enrich the understanding of model selection and implementation in the insurance sector.

Despite the superior predictive capacity of the GBM in this dataset, it is important to note that its main limitation is the lack of interpretable coefficients, which are easily obtained in the GLM and are crucial for inclusion in a traditional tariff structure. However, this limitation does not diminish the value of the GBM, but rather highlights the importance of a complementary approach. Machine learning models such as GBM can effectively complement GLM models. GBM can capture complex non-linear relationships between variables, which can be visualised using PDP or SHAP graphs. These insights can then be used to refine GLM modelling, for example by appropriately segmenting variables prior to modelling. This hybrid approach can significantly improve GLM modelling, resulting in a more accurate and robust tariff structure.

These findings have significant implications for the insurance sector, suggesting that the integration of machine learning techniques into pricing processes could lead to more accurate models and consequently more competitive pricing. Future studies could explore methods to increase the interpretability of GBM models or develop hybrid approaches that combine the benefits of GBM and GLM in a more integrated way.

Bibliography

[1] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf.

[2] P. Parodi. *Pricing in General Insurance*. 2014-10. ISBN: 9780429166471. DOI: 10.1201/b17525.

[3] M. N. M. A. K. Jain and P. J. Flynn. "Data clustering: A review". In: *ACM Computing Surveys*. 1999, pp. 264–323.

[4] N. Sai et al. "A Comparative Study of Distance-Based Clustering Algorithms in Fuzzy Failure Modes and Effects Analysis". In: 2023-05, pp. 605–624. ISBN: 978-981-99-1413-5. DOI: 10.1007/978-981-99-1414-2_45.

[5] S. Saqib et al. "Intelligent Dynamic Gesture Recognition Using CNN Empowered by Edit Distance". In: *Cmc -Tech Science Press-* 66 (2020-11), pp. 2061–2076. DOI: 10.32604/cmc.2020.013905.

[6] D. Bora. "Effect of Different Distance Measures on the Performance of K-Means Algorithm: An Experimental Study in Matlab". In: 5 (2014-05).

[7] G. Gan and E. Valdez. "Data Clustering with Actuarial Applications". In: *North American Actuarial Journal* 24 (2019-06), pp. 1–19. DOI: 10.1080/10920277.2019.1575242.

[8] E. Gan G. Valdez. "Data Clustering with Actuarial Applications". In: *North American Actuarial Journal*. 2008, pp. 168–186.

[9] G. S. Sebestyen. "Pattern recognition by an adaptive process of sample set construction". In: *IRE Transactions on Information Theory*. 1962, pp. 82–91.

[10] J. Macqueen. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. 1967, pp. 281–297.

[11] R. L. Thorndike. "Who belongs in the family?" In: *Psychometrika*. 1953, pp. 267–276.

[12] D. R. Cox. "Note on grouping." In: *Journal of the American Statistical Association*. 1957, pp. 543–547.

[13] W. D. Fisher. "On grouping for maximum homogeneity." In: *Journal of the American Statistical Association*. 1958, pp. 789–798.

[14] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. 2012-01. DOI: 10.1016/C2009-0-61819-5.

[15] b. "Review on determining number of Cluster in KMeans Clustering". In: *International Journal of Advance Research in Computer Science and Management Studies*. 2013, pp. 90–95.

[16] *Elbow Method for Optimal Cluster Number in K-Means*. URL: https://www.analyticsvidhya.com/blog/2021/01/in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/ (visited on 2024-05-20).

[17] O. e. a. Arbelaitz. "An extensive comparative study of cluster validity indices". In: (2012).

[18] S. Haberman and A. Renshaw. "Generalized Linear Models in Actuarial Work". In: *Journal of the Staple Inn Actuarial Society* 32 (1988-01). DOI: 10.1017/S2049929900010485.

[19] O. C. Njoku. "Decision Trees and Their Application for Classification and Regression Problems". Master's thesis. Missouri State University, 2019. URL: https://bearworks.missouristate.edu/theses/3406.

[20] J. Friedman. "Greedy Function Approximation: A Gradient Boosting Machine". In: *The Annals of Statistics* 29 (2000-11). DOI: 10.1214/aos/1013203451.

[21] P. McCullagh and J. Nelder. "Generalized Linear Models. Monographs on Statistics and Applied Probability No 37". In: *Chapman Hall, London* (1989-01). DOI: 10.1007/978-1-4899-3242-6.

[22] V. M. de Lizarduy Kostornichenko. "Comparative performance analysis between Grandient Boosting models and GLMs for non-life pricing". MA thesis. University Carlos III of Madrid, 2021.

[23] C. de Miranda Clemente. "A Refreshed Vision Of Non-Life Insurance Pricing". MA thesis. NOVA IMS, Master in Statistics and Information Management, 2022.