Stockholms universitet

A Comparison of Gradient Boosting Machines and Generalized Linear Models for Non-Life Insurance Pricing

Alexander Eriksson

Masteruppsats i försäkringsmatematik
Master Thesis in Actuarial Mathematics

Masteruppsats 2021:5
Försäkringsmatematik
Juni 2021

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Stockholm University

Mathematical Statistics
Stockholm University
Master Thesis **2021:5**
http://www.math.su.se

A Comparison of Gradient Boosting Machines and Generalized Linear Models for Non-Life Insurance Pricing

Alexander Eriksson*

June 2021

Abstract

The pricing of an insurance contract is a crucial area for risk assessment in the insurance business. Actuaries typically apply statistical methods to perform this task, which is known as rate making. This study puts focus on claim frequency modelling, an important part within non-life insurance pricing, with the application of gradient boosting machines (GBMs). GBMs are a family of statistical learning methods which have grown in popularity because of their strong performance in numerous disciplines. By using simulated data, inspired by real data provided by the insurance company Hedvig, we compare GBMs with generalized linear models (GLMs), which typically are applied by pricing actuaries. In insurance pricing, transparency and interpetability of models are key aspects for actual business application. Therefore, we also investigate available tools for model interpretation and use them for interpeting the GBMs. We further explore the possibility of creating an improved GLM based on the potential insights provided by the GBM. We show that the GBM outperforms the GLM in terms of both prediction accuracy and ranking of the claim frequency risk if we are not aware of present interaction effects. By using variable importance, partial dependence plots and Friedman's H-statistic we also show that we can gain an understanding of how the GBM models work and how we can extract insights from them. Further, the insights provided by the GBM guided us in the right direction towards creating GLMs with improved model performance.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: alexandereriksson88@gmail.com. Supervisor: Filip Lindskog.

ACKNOWLEDGEMENTS

First I would like to thank Anton Grip at Hedvig Försäkring for his valuable support and providing data for this thesis. I would also like to thank Stefan Arenbalk, Hao Huang and Robert Luciani at Foxrane AB for their guidance. In addition, I would like to thank my supervisor Filip Lindskog at Stockholm University for his valuable feedback and advice. Last but not least, I would like to thank my girlfriend Malin and the rest of my family for always supporting me.

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

insurance policy	a contract in which an insurance company agrees to compensate a policyholder if a specified uncertain future event adversely affects the policyholder
insurance company	an entity which provides insurance
policyholder	the buyer of an insurance policy, may be an individual or a business
exposure	the amount of time an insurance policy is in force, usually measured in years
claim	a request for payment reported by the policyholder to the insurer
loss cost	the sum of claim amounts an insurer must pay to the policyholder due to incurred claims
premium	the price a policyholder must pay for an insurance policy
risk premium	the part of the premium corresponding to the expected loss cost
claim frequency	the number of claims divided by the exposure
claim severity	the total claim amount divided by the number of claims
tariff	the set of rules used for computing the premium for an insurance policy
rate making	the determination of what premiums to charge for insurance policies

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

Statistical learning techniques have become increasingly popular in the last few years, following advances in computational processing and growing volumes of data. Since the insurance industry is highly dependent on making good data-driven decisions, these techniques are starting to get more attention.

Insurance companies play a key role in society by offering protection against financial losses. A fundamental task for insurance companies is how to price an insurance contract, which is also called rate making. Insurance companies set the price for an insurance contract before the actual cost of the contract is known. For this reason it is imperative that an insurer properly assesses the risks in its portfolio. Rate making is one of the key responsibilities and activities of actuaries. In this area, actuaries evaluate the potential losses to the insurance company and derive corresponding premiums. The data that is used for rate making is growing both in terms of the number of potential explanatory variables and amount of available data points.

In non-life insurance pricing, generalized linear models (GLMs) is the traditional tool of actuaries for rate making. GLMs are known to be very efficient for model fitting and also to provide easy to interpret results. However, applying GLMs require a detailed a priori knowledge about the structure of the data. If this structure is known, GLMs provide high predictive capability. It is common however that the structure is not completely known and that certain non-linear components are present in the data. These so called interactions need to be included manually in the process of fitting a GLM. To find and include these interactions can be a tedious analytical task for an actuary.

An alternative approach to GLMs, that provide more flexibility and that has been showing promising results, is gradient boosting, also called gradient boosting machines (GBMs). Compared to GLMs, GBMs do not require an a priori knowledge about the structure of the data. By applying gradient boosting, potential complex interactions among predictors can be captured automatically in the model fitting process. Thus, it is

of interest to investigate if a well defined GBM is able to predict more accurately than a GLM.

One of the drawbacks of machine learning (ML) techniques such as gradient boosting is that they are known for black-box effects, i.e. it can be difficult to interpret how the models actually work.

As mentioned in Henckaerts, Côté, Antonio, & Verbelen (2020), insurance pricing models are heavily regulated and should meet requirements with regard to transparency, fairness and solidarity before being deployed in practice. Henckaerts, Côté, Antonio, & Verbelen (2020) argue, with the support of Kaminski (2019), that by law individuals have the right to an explanation of the logic behind a decision according to the regime of "algorithmic accountability" that the European Union's General Data Protection Regulation (GDPR) establish. For this reason, pricing models need to be transparent and easy to communicate. The fairness perspective considers the idea that a policyholder should be charged a fair premium related to his or her risk profile to minimize adverse selection. Shortcomings in this regard could lead to underpricing (overpricing) of bad risks (good risks). The solidarity perspective reflects that an insurer has the social role of creating solidarity among policyholders. Therefore, the pricing should not be discriminatory or highly individualized. For all of these reasons it is of essence to be able to understand how the machine learning algorithm works, and how to interpret the model results.

In many circumstances it might be required that a pricing model has a GLM structure. Reasons for this could be e.g. regulatory requirements or system implementation considerations. However, by extracting the potential insights discovered by ML-algorithms one could potentially include these in a GLM. Thus, the GBM could be used to discover the most important variables and any interactions between them, to potentially be included in a GLM. Both Henckaerts, Côté, Antonio, & Verbelen (2020) and Yang, Qian, & Zou (2018) elaborate on this possibility, and Henckaerts, Côté, Antonio, & Verbelen (2020) explicitly leave for future work the building of a competitive GLM inspired by the GBM.

1.1 Previous work

There has been a number of previous studies of applying gradient boosting for insurance pricing. Guelman (2012) did one of the earlier studies in applying gradient boosting for non-life insurance pricing. In this study, GBM was compared with a conventional GLM in prediction of auto at-fault accident loss costs. Loss functions based on

Bernoulli deviance and squared error loss were used respectively for modeling frequency and severity. The results showed that the level of accuracy in out-of-sample prediction was higher for the GBM compared to the GLM. To interpret the results Guelman (2012) applied partial dependence plots and by analyzing the relative importance of predictors.

Wuthrich & Buser (2019) show how tree-based machine learning techniques, including gradient boosting machines, can be adapted and used for modeling the claim frequency in non-life insurance pricing.

A rigorous study was made by Henckaerts, Côté, Antonio, & Verbelen (2020) where a comparison was made of GLM, GAM, decision trees and GBM for predicting future costs associated with an insurance contract. In a case study the authors derive complete tariff plans based on models built for both frequency and severity of claims for a motor third party liability (MTPL) portfolio. Model performance was assessed both with out-of-sample deviance and with methods to measure model lift. The results of the study show that the GBMs outperform the classical GLMs. As motivated by the regime of algorithmic accountability the study also puts focus on ways of interpreting models. Like in Guelman (2012) variable importance and partial dependence plots are used and in addition, individual conditional expectations are assessed.

Further, Yang, Qian, & Zou (2018) proposed the TDBoost algorithm which uses gradient tree boosting based on the Tweedie distribution. In the study the suggested algorithm is applied to auto insurance claim data to fit Tweedie compound-Poisson models for modeling the pure premium. The results of modeling the pure premium with TDBoost are compared with a Tweedie GLM and a generalized additive model (GAM) by calculating the Gini index from the Lorenz curve on a held-out test data set. TDBoost showed more accurate premium predictions than the other models. The paper by Yang, Qian, & Zou (2018) also considers methods to deal with potential black box effects of applying gradient boosting. Also this study used variable importance and partial dependence plots to provide interpretable results. In Zhou, Qian, & Yang (2020) a variant of TDBoost, called EMTboost, is presented which is adapted specifically for handling extremely unbalanced claim data (zero-inflated).

Lee & Lin (2018) introduced delta boosting that is said be computationally more efficient compared to gradient boosting since it combines the regression and adjustments steps of the ordinary gradient boosting procedure. The model is applied to car insurance claims data and showed an improvement in both computing efficiency and predictive accuracy compared to gradient boosting.

An interesting work with regard to combining data science techniques with con-

ventional GLM in a non-life insurance pricing setting is discussed by Fujita, Tanaka, & Iwasawa (2020). The authors propose a new method, AGLM (Accurate GLM), which combines GLM with regularization techniques and applying a sort of feature transformation using discretization of numerical features and specific coding methodologies of dummy variables. In a case study based on car insurance data, the predictive accuracy of AGLM is compared with GLM, regularized GLM, GAM and GBM on held out test data set by calculating Poisson deviance. From this result AGLM performed best with GBM being a close second.

1.2 Objectives

Applying GBM for non-life insurance pricing has shown encouraging results from previous studies. However, the application of GBMs for non-life insurance pricing is still relatively new and further studies are required to investigate the potential of their practical use.

The objective of this thesis is to investigate and compare GBMs with traditional GLMs for the modeling of claim frequency. In order to put emphasis on algorithmic accountability, the need for models to be transparent and easy to explain, a further objective is to investigate available tools for model interpretation and to assess the interpretability of the results of gradient boosting in a non-life insurance pricing setting.

In addition, the thesis will expand upon the work of Henckaerts, Côté, Antonio, & Verbelen (2020) and Yang, Qian, & Zou (2018) by investigating the possibility of creating an improved GLM based on the potential insights provided by the GBM.

The study will be done as a simulation study of non-life home insurance data. The simulated data will be inspired by characteristics found in real data. The main advantages of this approach is that we will know the true underlying data generating function, which eases model comparisons, and at the same time the study will have some foundation in reality.

We also hope to provide some insights to the Swedish insurance company Hedvig who provided us with the data that worked as inspiration for the simulation study.

1.3 Disposition

In Chapter 2 we present a mathematical background to non-life insurance pricing, generalized linear models and gradient boosting machines, as well as tools for interpretation in machine learning and methods for evaluating model performance. We then outline

the methodology for the simulation study in Chapter 3. In Chapter 4 the results of the study are presented. Moreover, the results are discussed with regard to the objectives of the thesis in Chapter 5. Here we also highlight some limitations in the thesis and make some suggestions for future work. Finally, the thesis is concluded in Chapter 6.

CHAPTER 2
MATHEMATICAL BACKGROUND

2.1 Non-life insurance pricing

Before going into specific details about statistical modeling techniques for non-life insurance pricing, we will start by explaining some basic concepts.

A non-life insurance policy is a contract in which an insurance company agrees to compensate a policyholder if a specified uncertain future event (the insured event) adversely affects the policyholder. This means that if the policyholder is subject to an insured loss, the policyholder can file a claim with the insurer, which is a request to the insurer for payment of a sum of money according to the terms of the insurance policy. For this transfer of risk, the insurer demands a fee, called premium. A non-life insurance policy may cover property, people or legal liabilities. Some common examples of non-life insurance are home insurance or car insurance. In general, non-life insurance is any type of insurance other than life insurance. Since the insurer sets the price for an insurance contract before the actual cost of the contract is known, it is very important that the insurer has a good understanding of the risks in their portfolios. The loss of the insurance company, which is the sum of many smaller individual losses, is much more predictable than the individual losses themselves due to the law of large numbers. Hence, the overall loss to the insurer should be in line with its expected loss. For this reason, a general applied principle is that the premium should be based on the expected loss that is transferred from the policyholder to the insurer (Ohlsson & Johansson, 2010). However, different policies have different exposure to risk. For example, for a home insurance policy, living in an area with higher crime rates would generally mean higher expected losses due to theft compared to an area with lower crime rates. Expected losses vary between policies which is why statistical methods are key in non-life insurance pricing. Another reason

why it is important that an insurer charges a premium relative to the risk of a policy is to minimize the potential for adverse selection (Dionne, Gouriéroux, & Vanasse, 1999). If an insurer charges a too high premium relative to the market, policies will be lost to competitors with a more fair premium. On the contrary, if an insurer charges a too small premium relative to the market, the insurer would attract more high risk policies leading to eroding margins.

In order to price a non-life insurance contract, insurers predict the loss cost y for each policyholder based on observable characteristics x. Therefore, the insurer develops a predictive model f, which maps the rating factors x to the predicted loss cost $\hat{y}$ by setting $\hat{y} = f(x)$. A common approach for developing this predictive model is by separately modeling the frequency and severity of the claims, which gives us:

$$\textit{Risk premium} = \textit{Expected claim frequency} \times \textit{Expected claim severity} \tag{2.1}$$

Here, the risk premium (or also commonly referred to as the pure premium) is the same as the predicted (expected) loss cost. The rating factors in x usually are variables belonging to one of the following categories:

- **Properties of the policyholders**, e.g. the age of the policyholder
- **Properties of the insured objects**, e.g. the size of an apartment
- **Properties of the geographic region**, e.g. the population density of the residential area of the policyholder

As indicated by Equation (2.1), insurance companies often derive separate models for claim frequency and claim severity. In this thesis, the focus will be on claim frequency models.

2.2 Generalized linear models

Generalized linear models (GLMs) are models that generalize the classical linear regression model to models that do not necessarily have normally distributed response variables and where a function of the mean of the response is linear in the explanatory variables. The classical linear regression model has the form

$$y = X\beta + \epsilon$$

where y are outcomes of the random vector Y and is an $n \times 1$ vector of observations y_i where $i = 1, ..., n$; X is an $n \times p$ matrix of the observations of the explanatory variables; β is an $p \times 1$ vector of coefficients of the explanatory variables; and ϵ is an $n \times 1$ vector of the error terms. Nelder & Wedderburn (1972) outline the assumptions of this model in the following way:

- **Random component:** The components of Y are independent and normally distributed. The mean μ_i for each component are allowed to differ but they have common variance σ^2
- **Systematic component:** The p explanatory variables of the model are combined to give the linear predictor η, such that $\eta = X\beta$
- **Link function:** The relation between the random and systematic components is specified through the link function. In the linear model the link function is equal to the identity function, giving us $E[Y] = \mu = \eta$

GLMs consist of an extensive class of statistical models that include the linear models as a special case. Thus, the restriction of the linear model in terms of normality, constant variance and additivity of the effects are removed. In contrary, the response variable Y is assumed to be a member of the exponential family of distributions (Anderson et al., 2004). In GLMs the variance is permitted to vary with the mean of the distribution and the effect of explanatory variables on the response variable is assumed to be additive on a transformed scale. The GLM-analog to the above assumptions for the linear model can be summarised as follows:

- **Random component:** The components of Y are independent and comes from the exponential family of distributions.
- **Systematic component:** The p explanatory variables of the model are combined to give the linear predictor η, such that $\eta = X\beta$
- **Link function:** The relation between the random and systematic components is specified through the link function, g, which is differentiable and monotonic such that $E[Y] = \mu = g^{-1}(\eta)$

2.2.1 Exponential family of distributions

The exponential family of distributions is a broad class of distributions that have the same density form, including Normal, Poisson, gamma, inverse Gaussian and many more.

The exponential family of distributions is defined as:

$$f_{Y_i}(y_i;\theta_i,\phi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\right\} \tag{2.2}$$

where $a(\phi)$, $b(\theta_i)$ and $c(y_i,\phi)$ are pre-specified functions, θ_i is a parameter related to the mean and ϕ is a scale parameter related to the variance. Members of the exponential family of distributions have two basic properties:

1. the distribution is completely specified in terms of its mean and variance,
2. the variance of Y_i is a function of its mean

The mean and variance for this distribution are the following:

$$\begin{aligned} E[Y_i] &= b'(\theta_i) = \mu_i \\ Var(Y_i) &= b''(\theta_i)a(\phi) \end{aligned} \tag{2.3}$$

In (2.3) we have that $b''(\theta_i) = \mu_i' = V(\mu_i)$ which is known as the variance function. The variance function is important since within the exponential family of distributions a probability distribution is uniquely specified by its variance function (Ohlsson & Johansson, 2010).

2.2.2 Tweedie models

In non-life insurance it is often beneficial to use probability distributions that are closed with regard to scale transformations, also called *scale invariant* distributions (Ohlsson & Johansson, 2010). For example, if we measure claim frequency the modeling should not depend on if the frequency is measured in e.g. percent or basis points.

Tweedie models are the only models in the exponential family of distributions that are scale invariant. For some p these models are defined by having the variance function:

$$v(\mu) = \mu^p$$

The case $p = 0$ gives the normal distribution and $p = 2$ gives the gamma distribution, which is a popular choice for modeling claim severity. Choosing a p between 1 and 2 yields the Compound-Poisson distribution which can be used to model the risk premium directly.

However, the focus of this thesis will be the case $p = 1$. This gives $v(\mu) = \mu$, i.e. the Poisson distribution, which is a common choice for modeling insurance claims counts.

2.2.3 Link function

Assuming we have p coefficients of the explanatory variables of a GLM, $\beta_1, ..., \beta_p$, the link function links the mean to the linear structure of the model through

$$g(\mu_i) = \eta_i = \sum_{j=1}^{p} x_{ij}\beta_j \tag{2.4}$$

It is possible to choose different link functions, but some link functions are more suitable than others. A common choice is the *canonical link function*, which is defined as

$$\begin{aligned} g(\mu_i) &= b'(\theta_i)^{-1} \\ \Rightarrow g(\mu_i) &= \theta_i = \sum_{j=1}^{p} x_{ij}\beta_j \end{aligned} \tag{2.5}$$

For modeling the claim frequency with a Poisson assumption the log-link is a common choice, which we also will use throughout this thesis.

2.2.4 Maximum likelihood estimation of parameter coefficients

A common method for estimating the parameters in GLMs is by applying the maximum likelihood (ML) method. The idea of the ML method is to find the parameters for a model that maximizes the probability that the observed data comes from the chosen model.

The method can be explained as follows. Suppose that we have independent observations $y_1, ..., y_n$ where y_i is the outcome of the random variable Y_i with the probability density function $f(y_i|\theta)$, that depend on the unknown parameter $\theta \in \Theta \in \mathbb{R}^{\mathrm{k}}$ for some $k \in \mathbb{N}$. The likelihood for the observations is then given by

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} f(y_i|\theta) \tag{2.6}$$

Taking the log on both sides of this equation gives the log-likelihood

$$\log \mathcal{L}(\theta) = \prod_{i=1}^{n} \log f(y_i|\theta) \tag{2.7}$$

Since the log transformation is a monotone transformation it does not change the location of the maxima and it is easier to work with since many distributions used in non-life insurance involve exponentials. The parameter θ is then chosen such that it maximizes the probability of the observed data, which gives us

$$\hat{\theta} = \underset{\theta \in \Theta \in \mathbb{R}^{\mathrm{k}}}{\arg\max}\, \mathcal{L}(\theta) = \underset{\theta \in \Theta \in \mathbb{R}^{\mathrm{k}}}{\arg\max} \log \mathcal{L}(\theta) \tag{2.8}$$

To be able to apply this methodology to a regression problem we define the conditional response $Y|_{\boldsymbol{X}=\boldsymbol{x}}$. We then assume there exists a density function $f(\cdot|\theta(\boldsymbol{x}))$ depending on an unknown parameter function $\theta(\cdot)$ of the explanatory variables $\boldsymbol{x}$ with $Y|_{\boldsymbol{X}=\boldsymbol{x}} \sim f(\cdot|\theta(\boldsymbol{x}))$. If we have independent and identically distributed observations $(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)$, the likelihood of the observations $y_1, ..., y_n$, given the explanatory variables $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$, is then

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} f(y_i|\theta(\boldsymbol{x}_i)) \tag{2.9}$$

To find the optimal parameter $\hat{\theta}(\cdot)$ for (2.9) we solve for the parameter that maximizes the likelihood, similar to (2.8).

2.2.5 Model assumptions for claim frequency

Since claims frequency modeling is based on count data, it is typically assumed that the number of claims is Poisson distributed (Henckaerts, Côté, Antonio, & Verbelen, 2020). Assuming that N is a discrete random variable that follows a Poisson distribution with given expected frequency $\lambda > 0$ and exposure $v > 0$, our aim is to model the expected frequency $\lambda > 0$ in a way so that we can capture heterogeneous groups of policyholders in terms of risk by using the explanatory variables $\boldsymbol{x}$. The exposure v is often defined as risk years, which also will be used throughout this thesis. In terms of home insurance, the explanatory variables $\boldsymbol{x} = (x_1, ..., x_p)$ could be the examples of properties listed in the beginning of this chapter.

Hence, we have that the claim counts, N, given the explanatory variables $\boldsymbol{x}$, is distributed according to $N|_{\boldsymbol{X}=\boldsymbol{x}} \sim Pois(\lambda(\boldsymbol{x})v)$. Since we want to model the claim frequency

we now define the response

$$Y = \frac{N}{v} = f(\boldsymbol{X}) + \varepsilon$$

by following the notation in Zöchbauer, Wüthrich, & Buser (2016), which gives us the conditional response

$$Y|_{\boldsymbol{X}=\boldsymbol{x}} = \frac{N}{v}|_{\boldsymbol{X}=\boldsymbol{x}} = f(\boldsymbol{x}) + \varepsilon$$

where the residual term ε is determined by N. The conditional response is distributed according to

$$\mathbb{P}(Y = k/v|\boldsymbol{X} = \boldsymbol{x}) = \mathbb{P}(N = k|\boldsymbol{X} = \boldsymbol{x}) = e^{-\lambda(\boldsymbol{x})v}\frac{(\lambda(\boldsymbol{x})v)^k}{k!} \tag{2.10}$$

Thus, we have that the parameter function $\theta(\cdot)$ is given by $\theta(\boldsymbol{x}) = \lambda(\boldsymbol{x})v$. Further, we have that the regression function is given by

$$f(\boldsymbol{x}) = \mathbb{E}(Y|\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{v}\mathbb{E}(N|\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{v}\theta(\boldsymbol{x})$$

In this thesis we will consider the special case of the Poisson model with a multiplicative regression structure. Using this regression structure means that we will apply the log-link function, as mentioned in Section 2.2.3. If we assume $X \subset \mathbb{R}^{\mathrm{p}}$ and that the regression function $f(\cdot)$ is determined by a parameter vector $\boldsymbol{\beta} = (\beta_0, \beta_1, ..., \beta_p)' \in \mathbb{R}^{\mathrm{p}+1}$ we have for $\boldsymbol{x}$ that

$$\log f(\boldsymbol{x}) = \log \lambda(\boldsymbol{x}) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$$

Now, given the observations $(N_1, \boldsymbol{x}_1, v_1), (N_2, \boldsymbol{x}_2, v_2), ..., (N_n, \boldsymbol{x}_n, v_n)$ for insurance policy $i = 1, ..., n$ with explanatory variables $\boldsymbol{x}_i$ and exposure v_i assuming that the observations are independent with N_i being Poisson distributed with expected frequency $\lambda(\boldsymbol{x}_i)$ we get the joint log-likelihood function

$$\ell_{\boldsymbol{N}}(\boldsymbol{\beta}) = \sum_{i=1}^{n} -v_i e^{(\beta_0+\beta_1 x_{i1}+...+\beta_p x_{ip})} + N_i(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} + \log v_i) - \log(N_i!)$$

To estimate the parameters we can apply the ML method to find the MLE $\hat{\boldsymbol{\beta}}$ by considering the solution of

$$\frac{\partial}{\partial \boldsymbol{\beta}}\ell_{\boldsymbol{N}}(\boldsymbol{\beta}) = 0$$

Further derivations can be found in Wuthrich & Buser (2019).

2.3 Tree-based methods

Since we will apply tree-based gradient boosting in this thesis, we will start this section by introducing some basic concepts related to decision trees. Thereafter, we continue by explaining the specifics of the gradient boosting algorithm. The foundation for these parts is based on the book *The Elements of Statistical Learning* by Hastie, Tibshirani, & Friedman (2009). We will finish this section by discussing the choice of loss function when modeling the claim frequency.

2.3.1 Decision trees

Decision trees are non-linear models that divide data based on yes-no questions and predict the same value for each observation belonging to a certain segment. Decision trees can be used for both regression problems or classification problems, then called regression trees and classification trees respectively. A popular method for constructing decision trees is the Classification and Regression Tree (CART) algorithm, which now will be discussed.

Assume we want to predict a response variable $Y \in \mathbb{R}$, that is a function of $\boldsymbol{X} \in \mathbb{R}^{\mathrm{p}}$ of p predictors. Now, let the predictor space R be the set of possible values for the p predictors $x_1, x_2..., x_p$. The method then works by dividing the predictor space R into J non-overlapping regions $R_1, R_2, ..., R_j$ where each region contain homogeneous groups of observations.

The decision tree model is defined as follows:

$$f(\boldsymbol{x}) = \sum_{j=1}^{J} \hat{y}_{R_j} I(\boldsymbol{x} \in R_j) \tag{2.11}$$

In (2.11) I is an indicator function that equals one for observations belonging to the J:th region and zero otherwise. Since the regions do not overlap the function I is one for exactly one region for each $\boldsymbol{x}$, which means that the tree makes a constant prediction $\hat{y}_{R_j}$ in the whole region R_j.

In the case of regression, the fitted response $\hat{y}_{R_j}$ in the j:th region is given by the average of the training observations in this region. For classification $\hat{y}_{R_j}$ is instead given as the mode of the j:th region.

One problem with growing trees is that there are extremely many ways in which one could form the regions. It would be computationally unfeasible to consider every possible partition of the predictor space. For this reason CART uses a greedy approach

called *recursive binary splitting*, which we now will explain.

First consider a node d, a splitting variable x_v where $v \in (1, 2..., p)$ from the predictor space R and a cut-off c. CART then splits d into a left node, $d_L = \{R \mid x_v \leq c\}$ and right node $d_R = \{R \mid x_v > c\}$ such that $d = d_L \cup d_R$. The result is a split into two nodes (a left node and right node) where one of them contain the observations for which $x_v \leq c$ and the other observations where $x_v > c$. This procedure continues recursively until a stopping criterion is fulfilled, e.g. the maximum depth of the tree or a minimum number of observations in a node. In general, for observations $i = 1, ..., n$ in the training data the CART algorithm searches for x_v and c such that some loss function $L(\cdot)$ is minimized given the two daughter nodes according to:

$$\arg\min_{v,c} \left[\sum_{i:\boldsymbol{x}_i \in d_L(v,c)} L(y_i, \hat{y}_{d_L}) + \sum_{i:\boldsymbol{x}_i \in d_R(v,c)} L(y_i, \hat{y}_{d_R}) \right] \tag{2.12}$$

It is common to use squared error loss as the loss function, but further in this chapter we present a loss function more suitable for claim frequency.

A question is how large tree one should grow. A very large tree could end up overfitting the data, while a small tree potentially does not capture important structures. Tree size is a tuning parameter which determines the complexity of the model and should be chosen from the data (Hastie, Tibshirani, & Friedman, 2009).

Since the focus of this thesis is the modeling of claims frequency, our primary interest are regression trees.

2.3.2 Gradient boosting machines

A decision tree model is easy to interpret and visualize. However, they often suffer from high variance and can be very sensitive to changes in the training data (Hastie, Tibshirani, & Friedman, 2009). To deal with this problem so called ensemble methods can be used in which multiple weaker models are combined to form a more powerful model. There are many different ensemble techniques but GBMs are among the most popular. In the ML community GBMs are known for their good results on basically any given problem and often win ML competitions. The **gradient boosting machine** was introduced by Friedman (2001) and is an iterative statistical method that combines many weak models, also called **learners**, into a strong predictor. In a predictive learning problem we have a response variable y and a set of explanatory variables $\boldsymbol{x} = (x_1, ..., x_p)$. By using a training sample of observations $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_n, y_n)$ the goal is to get an estimate $\hat{f}(\boldsymbol{x})$ of

the function $f^*(\boldsymbol{x})$ which maps $\boldsymbol{x}$ to y and that minimizes the expected value of some specified loss function $L(y, f(\boldsymbol{x}))$ over the joint distribution of all $(\boldsymbol{x}, y)$, according to

$$f^* = \arg\min_f E[L(y, f(\boldsymbol{x}))] \tag{2.13}$$

Since we only have the training sample at our hands, we obtain the approximation to (2.13) as

$$\hat{f} = \arg\min_f \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\boldsymbol{x}_i)) \tag{2.14}$$

A starting point for solving (2.14) is to restrict $f(\boldsymbol{x})$ to be a member of a parameterized class of functions, instead making the optimization problem about finding the optimal parameters for $f(\boldsymbol{x})$. Friedman (2001) introduced $f(\boldsymbol{x})$ as additive expansions of the form,

$$f(\boldsymbol{x}) = \sum_{m=1}^{M} \beta_m h(\boldsymbol{x}; \boldsymbol{a}_m) \tag{2.15}$$

where the function $h(\boldsymbol{x}; \boldsymbol{a})$ in (2.15) usually is a simple parameterized function of the input variables $\boldsymbol{x}$ with parameters $\boldsymbol{a}$. The procedure works sequentially and starts by making an initial guess $\hat{f}_0(\boldsymbol{x})$, and for each step, or boost, m, the pseudo residuals are used to to find regions of the predictor space where the model fit is poor in order to improve the fit in a direction of better performance. The pseudo residual $\boldsymbol{r}_{i,m}$ for observation i in iteration m is computed as the negative gradient of the loss function according to:

$$\boldsymbol{r}_{i,m} = -\left[\frac{\partial L(y_i, f(\boldsymbol{x}_i))}{\partial f(\boldsymbol{x}_i)}\right] \tag{2.16}$$

which is evaluated at the current model fit. This methodology is referred to as gradient descent which gives a lower loss for every iteration until convergence. Next we fit the base learner $h(\boldsymbol{x}; \boldsymbol{a}_m)$ to the pseudo residuals $\boldsymbol{r}_{i,m}$ using the training data $\{(\boldsymbol{x}, \boldsymbol{r}_{i,m})\}_{i=1}^{n}\}$ by least squares:

$$\boldsymbol{a}_m = \arg\min_{\boldsymbol{a},\beta} \sum_{i=1}^{n} [\boldsymbol{r}_{i,m} - \beta h(\boldsymbol{x}_i; \boldsymbol{a})]^2$$

Then by performing line search a step length ρ_m is computed by solving the following optimization problem:

$$\rho_m = \arg\min_{\rho} \sum_{i=1}^{n} L(y_i, f_{m-1}(\boldsymbol{x}_i) + \rho h(\boldsymbol{x}_i; \boldsymbol{a}_m)) \tag{2.17}$$

which is the step length that yields a maximum decrease of the empirical loss.

We can then update the model according to:

$$f_m(\boldsymbol{x}) = f_{m-1}(\boldsymbol{x}) + \tau\rho_m h(\boldsymbol{x}; \boldsymbol{a}_m) \tag{2.18}$$

where $0 < \tau \leq 1$ is a shrinkage parameter controlling the learning speed of the model. A lower τ usually results in better performance but also increases computation time because more iterations are needed for convergence to a good solution (Henckaerts, Côté, Antonio, & Verbelen, 2020).

Now we will consider the case when using regression trees as base learners, which is the focus of this thesis. In this case the base learner is given by:

$$h(\boldsymbol{x}; \{b_j, R_j\}_1^J) = \sum_{j=1}^{J} b_j I(\boldsymbol{x} \in R_j)$$

where the parameters now are the coefficients $\{b_j\}_1^J$ and the values that define the boundaries of the regions $\{R_j\}_1^J$, which are the splitting variables and splitting points that defines the tree. Here the model updates in (2.18) becomes

$$f_m(\boldsymbol{x}) = f_{m-1}(\boldsymbol{x}) + \tau\rho_m \sum_{j=1}^{J_m} b_{jm} I(\boldsymbol{x} \in R_{jm}) \tag{2.19}$$

where $\{R_{jm}\}_1^{J_m}$ are the regions that are defined by the terminal nodes of the tree at the m:th iteration. These are used to predict the pseudo residuals $\boldsymbol{r}_{i,m}$ in (2.16) by least squares and $\{b_{jm}\}$ are the respective fitted coefficients. The factor ρ_m is the line search solution according to Equation (2.17).

The model updates can be expressed as

$$f_m(\boldsymbol{x}) = f_{m-1}(\boldsymbol{x}) + \tau \sum_{j=1}^{J_m} \gamma_{jm} I(\boldsymbol{x} \in R_{jm}) \tag{2.20}$$

where $\gamma_{jm} = \rho_m b_{jm}$. Now, one can view (2.20) as adding J_m separate basis functions at each step instead of a single additive one as in Equation (2.19). It is possible to further improve the model fit by using the optimal coefficients for each of these separate basis

functions in Equation (2.20), which are the solution to

$$\{\gamma_{jm}\}_1^J = \arg\min_{\{\gamma_j\}_1^J} \sum_{i=1}^{n} L\left(y_i, f_{m-1}(\boldsymbol{x}_i) + \sum_{j=1}^{J} \gamma_j I(\boldsymbol{x} \in R_{jm})\right) \tag{2.21}$$

Since the regions are disjoint, this reduces to

$$\gamma_{jm} = \arg\min_{\gamma} \sum_{\boldsymbol{x}_i \in R_{jm}} L(y_i, f_{m-1}(\boldsymbol{x}_i) + \gamma) \tag{2.22}$$

A further development of the gradient boosting procedure was done by Friedman (2002), called stochastic gradient boosting, which introduces randomness into the procedure. The idea is to randomly select a subsample of size $\delta \cdot n$ of the data in each iteration m for the model update, which was shown to improve both predictive accuracy and execution time when $\delta < 1$. This procedure, specifically based on regression trees, will be the model for competing with the GLM in this thesis. Algorithm 1 provides a summary of this procedure in pseudo code.

Algorithm 1: Stochastic gradient tree boosting

start by finding the optimal constant model: $f_0(\boldsymbol{x}) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$;

for *m=1,...,M* **do**

  randomly subsample data of size $\delta \cdot n$ without replacement from the training data;

  for *i=1,...,*$\delta \cdot n$ **do**

    $\boldsymbol{r}_{i,m} = -\left[\frac{\partial L(y_i, f(\boldsymbol{x}_i))}{\partial f(\boldsymbol{x}_i)}\right]$;

    fit a tree of depth d to the pseudo residuals $\boldsymbol{r}_{i,m}$ which gives regions $R_{j,m}$ for $j = 1, ..., J_m$;

  end for

  for *j=1,...,*J_m **do**

    $\gamma_{jm} = \arg\min_{\gamma} \sum_{\boldsymbol{x}_i \in R_{jm}} L(y_i, f_{m-1}(\boldsymbol{x}_i) + \gamma)$;

    update $f_m(\boldsymbol{x}) = f_{m-1}(\boldsymbol{x}) + \tau \sum_{j=1}^{J_m} \gamma_{jm} I(\boldsymbol{x} \in R_{jm})$;

  end for

  $f_{gbm}(\boldsymbol{x}) = f_M(\boldsymbol{x})$;

end for

2.3.3 Loss function for claim frequency

For applying machine learning algorithms like decision trees or GBMs we need to specify which loss function to minimize when training the model. The typical loss function for regression problems is the squared error loss:

$$L(y_i, f(\boldsymbol{x}_i)) = (y_i - f(\boldsymbol{x}_i))^2$$

where y_i is the observed response and $f(\boldsymbol{x}_i)$ is the prediction evaluated at $\boldsymbol{x}_i$. However, squared error loss is only appropriate when the data is normally distributed. Claim frequency on the other hand is not normally distributed. Since claim frequency typically is assumed to be Poisson distributed, Wuthrich & Buser (2019) suggest Poisson deviance as an appropriate loss function. For a homogeneous portfolio with the same expected frequency $\lambda > 0$ this can be defined as:

$$\begin{aligned} D^*(\boldsymbol{N}, \lambda) &= 2(\ell_{\boldsymbol{N}}(\boldsymbol{N}) - \ell_{\boldsymbol{N}}(\lambda)) \\ &= 2\sum_{i=1}^{n} -N_i + N_i \log N_i + \lambda v_i - N_i \log(\lambda v_i) \\ &= \sum_{i=1}^{n} 2\ N_i \left[\frac{\lambda v_i}{N_i} - 1 - \log\left(\frac{\lambda v_i}{N_i} \right) \right] \end{aligned} \tag{2.23}$$

By considering the case where we instead have the regression function $f(\boldsymbol{x}) = \lambda(\boldsymbol{x})$ for the claim frequency, we can adjust (2.23) to get

$$\begin{aligned} D^*(\boldsymbol{N}, \lambda(\boldsymbol{x})) &= 2(\ell_{\boldsymbol{N}}(\boldsymbol{N}) - \ell_{\boldsymbol{N}}(\lambda(\boldsymbol{x}))) \\ &= 2\sum_{i=1}^{n} -N_i + N_i \log N_i + \lambda(\boldsymbol{x}_i) v_i - N_i \log(\lambda(\boldsymbol{x}_i) v_i) \\ &= \sum_{i=1}^{n} 2\ N_i \left[\frac{\lambda(\boldsymbol{x}_i) v_i}{N_i} - 1 - \log\left(\frac{\lambda(\boldsymbol{x}_i) v_i}{N_i} \right) \right] \end{aligned} \tag{2.24}$$

This loss function will be used throughout this thesis for building the GBMs.

2.4 Tools for interpration in machine learning

Single decision trees are easy to interpret since the model can be completely represented and visualized with a two-dimensional graphic. However, combinations of multiple trees such as the GBM do not have this feature, resulting in their black-box nature. To

be able to open up this black-box and gain understanding of how a model like the GBM works, there are several tools and methods available. This section will present the tools we will use in this thesis to interpret how the GBM models work.

2.4.1 Variable importance

Variable importance was introduced by Breiman, Friedman, Stone, & Olshen (1984) and is a measure of how important the explanatory variables are in predicting the response. For a specific explanatory variable x_ℓ, $\ell \in \{1, ..., p\}$, in a decision tree m the variable importance is given by:

$$\mathcal{I}_\ell(m) = \sum_{j=1}^{J-1} I(v(j) = \ell) \, (\Delta L)_j \tag{2.25}$$

Thus, the importance is measured by taking the sum of the improvements in the loss function L over all the internal nodes $J-1$ for which the variable x_ℓ was used as the splitting variable. More important variables would accumulate larger improvements in the loss function over the splits than less important ones. In order to understand the relative contribution of each variable, we normalize the importance measures so they sum to 100 %. Now, this idea can easily be applied to ensemble techniques such as GBMs by averaging the importance of variable x_ℓ over the different trees included in the ensemble according to:

$$\mathcal{I}_\ell = \frac{1}{M} \sum_{m=1}^{M} \mathcal{I}_\ell(m) \tag{2.26}$$

2.4.2 Partial dependence plots

After the most important variables have been identified, it is meaningful to understand their effect on the response. Partial dependence plots (PDPs) show the marginal effect of a variable on the predictions obtained from a model (Hastie, Tibshirani, & Friedman, 2009). This means that we calculate the predictions for a specific variable x_ℓ while averaging over the values of other variables $\boldsymbol{x}_C$ according to:

$$\bar{f}_\ell(x_\ell) = \frac{1}{n} \sum_{i=1}^{n} f_{model}(x_\ell, \boldsymbol{x}_{i,C}) \tag{2.27}$$

where C is the complement set to ℓ such that $\ell \cup C = \{1, ..., p\}$; $\boldsymbol{x}_{i,C}$ are the values of the other variables for observation i; and n is the number of observations in the training

data.

It is important to note that PDPs measure the effect of x_ℓ on $f(\boldsymbol{x})$ after accounting for the average effects of other variables $\boldsymbol{x}_c$ on $f(\boldsymbol{x})$. For this reason potential interaction effects between x_ℓ and another variable in $\boldsymbol{x}_c$ might obscure the effect.

2.4.3 Individual conditional expectation

One way to handle the potential problem with PDPs is to look at individual conditional expectations, which instead consider the effect of a variable on the predictions on the level of individual observations (Henckaerts, Côté, Antonio, & Verbelen, 2020). This is given by:

$$\bar{f}_{\ell,i}(x_\ell) = f_{model}(x_\ell, \boldsymbol{x}_{i,C}) \tag{2.28}$$

By calculating the individual conditional expectation for each observation i it will be possible to discover potential interaction effects if groups of observations show different effect patterns.

2.4.4 Friedman's H-statistic

In order to identify potential interaction signals in the data we will analyze Friedman's H-statistic. The H-statistic was introduced by Friedman, Popescu, & others (2008) and estimates the strength of interactions by measuring how much of the prediction variance that stems from the interaction effect. In this thesis, we will restrict ourselves to two-way interactions but the H-statistic could be applied to any number of variables. Let $\bar{f}_k(x_k)$ and $\bar{f}_\ell(x_\ell)$ be the one-way partial dependence of the variables x_k and x_ℓ, and $\bar{f}_{k\ell}(x_k, x_\ell)$ the two-way partial dependence defined similarly by Equation (2.27). The H-statistic can then be defined as

$$H^2_{k\ell} = \frac{\sum_{i=1}^n \{\bar{f}_{k\ell}(x_k^{(i)}, x_\ell^{(i)}) - \bar{f}_k(x_k^{(i)}) - \bar{f}_\ell(x_\ell^{(i)})\}^2}{\sum_{i=1}^n \{\bar{f}^2_{k\ell}(x_k^{(i)}, x_\ell^{(i)})\}} \tag{2.29}$$

where $x_k^{(i)}$ is the observed value x_k for observation i. Equation (2.29) represents the variance of the interaction divided by the total variance. Thus, this measure is between zero and one, where zero indicates no interaction and one would mean that the effect of x_k and x_ℓ on the predicted response is only influenced by the interaction.

2.5 Model performance

To be able to compare the model performance of the GLMs and GBMs, we need to introduce relevant performance measures and discuss how they can be applied. To make the model performance assessment more rigorous, we will assess the models both from a statistical perspective and from the perspective of using them in practice, i.e. as a part in building a pricing tariff. First we will explain the *bias-variance tradeoff* to gain understanding of why model performance assessment is important. Then follows a presentation of our selected cross-validation technique for model training and assessment and how we will assess the statistical performance of the models as well as their performance in a pricing tariff.

2.5.1 The bias-variance tradeoff

An important concept in statistical modeling is the so called *bias-variance tradeoff*. But before we discuss this tradeoff, we need to define what bias and variance represent. Bias refers to the error produced by a model because of underfitting the training data. The reasons for this might be e.g. because the chosen model is too simple or that not enough explanatory variables are included in the model. If a model is underfitting, this means that we have high bias. Variance instead refers to how much a model is overfitting the training data. The reasons for this could be e.g. because the chosen model is too complex or that too many explanatory variables are included in the model. If a model is overfitting, this means that we have high variance. In an ideal situation, we would like to select a model that accurately captures patterns in the training data and at the same time generalizes well on new data. However, it is usually impossible to do both at the same time. If we choose to lower bias, we typically increase variance and if we choose to lower variance, we typically increase bias. To select a good model, we need to balance this tradeoff between bias and variance, without overfitting or underfitting the data. This tradeoff is illustrated in Figure 2.1.

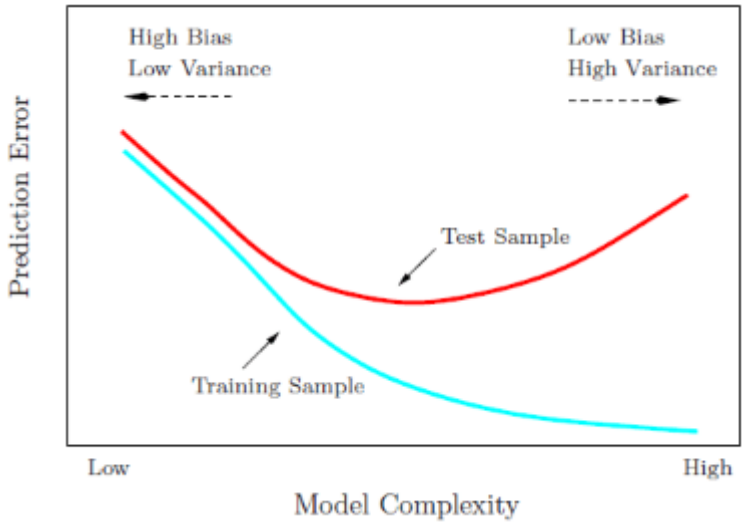

Figure 2.1: The bias-variance tradeoff

Different types of models tend to either have high bias and low variance or low bias and high variance. For example, linear regression models tend to have high bias, since they assume a simple linear relationship between the explanatory variables and the response, and low variance, since estimates likely will not change significantly from one training dataset to another. Complex non-linear models, on the other hand, tend to have low bias, since they do not assume any specific relationship between the explanatory variables and the response, and high variance, since model estimates can change significantly from one training dataset to another. Hence, an important thing to consider for this thesis is that the GBM, being a non-linear model, will be more likely overfit the data compared to the GLM which is a linear model. The GLM, on the other hand, will be more likely to underfit the data compared to the GBM.

2.5.2 Cross-validation

For validating machine learning models it is common to apply so called nested cross-validation, since it can deal both with the selection of the best set of hyperparameters and error estimation. There are many variants of nested cross validation, but for validating the models in this thesis we have chosen to adopt an approach inspired by Henckaerts, Côté, Antonio, & Verbelen (2020). The method works as follows. We start by partitioning a dataset $\mathcal{D}$ into K disjoint, equally sized and stratified sets $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$ ordered on claim frequency. By performing stratification based on claim frequency, we get a similar distribution of the claim frequency in each of the K datasets. These K subsets are iterated over in an outer for loop, where for each iteration k the k:th subset in

$\mathcal{D}_k \in \{\mathcal{D}_1, ..., \mathcal{D}_K\}$ is left out as test set $\mathcal{D}_k$. Next, for every hyperparameter combination, the remaining data $\mathcal{D} \setminus \mathcal{D}_k$ is iterated over in an inner for loop, where for each iteration l the l:th subset in $\mathcal{D}_l \in \{\mathcal{D}_1, ..., \mathcal{D}_K\} \setminus \mathcal{D}_k$ is left out as validation set $\mathcal{D}_l$ and a model is trained on the data $\mathcal{D} \setminus \{\mathcal{D}_k, \mathcal{D}_l\}$. Thus, in the inner for loop $K-1$ cross validation is performed for every hyperparameter combination and the cross validation error, applying loss function L, computed by averaging the error on the validation data sets. The optimal hyperparameters are those that minimize the cross validation error. The optimal hyperparameters are then used to train a model on data $\mathcal{D} \setminus \mathcal{D}_k$ and the model performance evaluated on the test data $\mathcal{D}_k$.

More specifically, we will apply this algorithm for cross-validation where $K = 6$, giving us six data folds. The procedure that is applied for fitting the GBMs in this thesis is summarised in Algorithm 2. Since we will be modelling claim frequency, the loss function given by Equation (2.24) will be applied to fit the models and compute the validation error and test error.

Algorithm 2: nested stratified 6-fold cross-validation

Input: the model type and the respective grid of hyperparameters (*hypgrid*)
Output: optimal hyperparameters and computed performance for each of the six data folds.

partition the data $\mathcal{D}$ into 6 disjoint stratified and equally sized data sets $\mathcal{D}_1, ..., \mathcal{D}_6$;
for *k=1,...,6* **do**
  foreach *combination of hyperparameters in* ***hypgrid*** **do**
    leave out $\mathcal{D}_k$ as a test data set;
    for $l \in \{1, ..., 6\} \setminus k$ **do**
      train a model f_{kl} on $\mathcal{D} \setminus \{\mathcal{D}_k, \mathcal{D}_l\}$;
      compute model performance on $\mathcal{D}_l$ by applying the loss function $L(\cdot, \cdot)$;
      compute validation error $valerr_{kl} = \frac{1}{|\mathcal{D}_l|} \sum_{i \in \mathcal{D}_l} L(y_i, f_{kl}(\boldsymbol{x}_i))$;
    end for
    compute validation error $valerr_k = \frac{1}{5} \sum_{l \in \{1,...,6\} \setminus k} valerr_{kl}$;
  end foreach
  the optimal parameters in *hypgrid* are the ones that minimize $valerr_k$;
  we train a model f_k on $\mathcal{D} \setminus \mathcal{D}_k$ by using the optimal parameters;
  compute the model performance on $\mathcal{D}_k$ by applying the loss function $L(\cdot, \cdot)$;
  $testerr_k = \frac{1}{|\mathcal{D}_k|} \sum_{i \in \mathcal{D}_k} L(y_i, f_k(\boldsymbol{x}_i))$;
end for

2.5.3 Generalization loss

In order to assess the quality of an estimated model we can estimate the generalization loss (Wuthrich & Buser, 2019). Hastie, Tibshirani, & Friedman (2009) express the generalization loss, or generalization error, as the prediction error over an independent test sample according to

$$Err_{\mathcal{T}} = \mathbb{E}\left[L(Y, \hat{f}(\boldsymbol{x}))|\mathcal{T}\right] \tag{2.30}$$

with response Y, explanatory variable $\boldsymbol{x}$, prediction function $\hat{f}(\cdot)$ and loss function L. The training data set $\mathcal{T}$ is fixed in Equation (2.30) and the test error thus refers to the error for this specific training data set.

A related measure is the expected prediction error

$$Err = \mathbb{E}\left[L(Y, \hat{f}(\boldsymbol{x}))\right] = \mathbb{E}\left[Err_{\mathcal{T}}\right] \tag{2.31}$$

Here we average over everything that is random, which includes the randomness of choosing the training data $\mathcal{T}$ that generates $\hat{f}$.

Since we will be working with the Poisson deviance loss for modeling the claim frequency, according to Equation (2.24), we have in our specific case that the so called Poisson deviance generalization loss is

$$Err = \mathbb{E}\left[D^*(\boldsymbol{N}, \hat{\lambda(\boldsymbol{x})})\right] = 2\mathbb{E}\left[\hat{\lambda(\boldsymbol{x})}v - \boldsymbol{N} - \boldsymbol{N}\log(\lambda(\boldsymbol{x})\hat{v}/\boldsymbol{N})\right] \tag{2.32}$$

Moreover, according to Wuthrich & Buser (2019), we have that we can express the Poisson deviance generalization loss as

$$Err = \mathbb{E}[D^*(\boldsymbol{N}, \hat{\lambda(\boldsymbol{x})})] = \mathbb{E}\left[D^*(\boldsymbol{N}, \lambda(\boldsymbol{x}))\right] + \mathcal{E}(\hat{\lambda(\boldsymbol{x})}, \lambda(\boldsymbol{x})) \tag{2.33}$$

where $\mathcal{E}(\hat{\lambda(\boldsymbol{x})}, \lambda(\boldsymbol{x}))$ is the estimation loss defined by

$$\mathcal{E}(\hat{\lambda(\boldsymbol{x})}, \lambda(\boldsymbol{x})) = 2v\left[\mathbb{E}[\hat{\lambda(\boldsymbol{x})}] - \lambda(\boldsymbol{x}) - \lambda(\boldsymbol{x})\mathbb{E}[\log(\hat{\lambda(\boldsymbol{x})}/\lambda(\boldsymbol{x}))]\right] \geq 0 \tag{2.34}$$

This estimation loss quantifies the bias and the estimation variance simultaneously.

Now we consider the specific case of computing the Poisson deviance generalization loss and estimation loss for our six data folds. For data fold k a model is trained on the data $\mathcal{D} \setminus \mathcal{D}_k$, with $\mathcal{D}_k$ left out as test data. We denote training data $\mathcal{D} \setminus \mathcal{D}_k$ by $\mathcal{D}^{train_k}$ and corresponding test data $\mathcal{D}_k$ as $\mathcal{D}^{test_k}$.

By using the training data $\mathcal{D}^{train_k}$ we can compute the prediction function $f(\hat{\boldsymbol{x}})^{train_k} = \lambda(\hat{\boldsymbol{x}})^{train_k}$ and then use the test data $\mathcal{D}^{test_k}$ to estimate the out-of-sample generalization loss for data fold k of the claim frequency empirically according to

$$Err^{freq}_{\mathcal{D}^{train_k}} = \hat{\mathbb{E}}\left[D^*(\boldsymbol{N}, \lambda(\hat{\boldsymbol{x}})^{train_k})|\mathcal{D}^{train_k}\right]$$
$$= \frac{1}{|\mathcal{D}^{test_k}|}\sum_{i\in\mathcal{D}^{test_k}} 2\ N_i\left[\frac{v_i\lambda(\hat{\boldsymbol{x}}_i)^{train_k}}{N_i} - 1 - \log\left(\frac{v_i\lambda(\hat{\boldsymbol{x}}_i)^{train_k}}{N_i}\right)\right] \tag{2.35}$$

which can also be referred to as the mean out-of-sample Poisson deviance loss, given training data $\mathcal{D}^{train_i}$.

Similarly, we can compute the out-of-sample estimation loss for data fold k according to

$$\mathcal{E}(\lambda(\hat{\boldsymbol{x}})^{train_k}, \lambda(\boldsymbol{x})) =$$
$$\frac{1}{|\mathcal{D}^{test_k}|}\sum_{i\in\mathcal{D}^{test_k}} 2\ v_i\left[\lambda(\hat{\boldsymbol{x}}_i)^{train_k} - \lambda(\boldsymbol{x}_i) - \lambda(\boldsymbol{x}_i)\log\left(\frac{\lambda(\hat{\boldsymbol{x}}_i)^{train_k}}{\lambda(\boldsymbol{x}_i)}\right)\right] \tag{2.36}$$

To approximate the overall generalization loss and estimation loss we can compute the average generalization loss and estimation loss over the six data folds we will use in our study. For the generalization loss we then have

$$Err^{freq} = \frac{1}{n}\sum_{k=1}^{6}\left\{\sum_{i\in\mathcal{D}^{test_k}} 2\ N_i\left[\frac{v_i\lambda(\hat{\boldsymbol{x}})^{train_k}}{N_i} - 1 - \log\left(\frac{v_i\lambda(\hat{\boldsymbol{x}})^{train_k}}{N_i}\right)\right]\right\} \tag{2.37}$$

and for the estimation loss we have

$$\mathcal{E}(\lambda(\hat{\boldsymbol{x}}), \lambda(\boldsymbol{x})) =$$
$$\frac{1}{n}\sum_{k=1}^{6}\left\{\sum_{i\in\mathcal{D}^{test_k}} 2\ v_i\left[\lambda(\hat{\boldsymbol{x}}_i)^{train_k} - \lambda(\boldsymbol{x}_i) - \lambda(\boldsymbol{x}_i)\log\left(\frac{\lambda(\hat{\boldsymbol{x}}_i)^{train_k}}{\lambda(\boldsymbol{x}_i)}\right)\right]\right\} \tag{2.38}$$

To assess model performance in practice one would typically use generalization losses. However, since we will be working with simulated data in this thesis, we will be in the special situation of knowing the true frequency function and thus be able to quantify the estimation loss. For this reason, the estimation loss according to Equation (2.38) will be our main performance measure for model assessment. We will also assess the generalization loss according to Equation (2.37) to see if we draw the right conclusions based on the cross-validation analysis.

2.5.4 Model lift

Besides evaluating the statistical performance of the models we will evaluate the overall ability of the models to rank the claim frequency risk and prevent adverse selection. This is an important input for making the business decision of choosing between different models since it translates into the actual economic value of the models. In terms of rate making in insurance, economic value can be evaluated by assessing model lift (Goldburd, Khare, & Tevet, 2016). Model lift is a relative measure, which means that two or more competing models are needed as input. Hence, when we talk about model lift we mean the lift of one model over another.

Model lift should always be measured on holdout data to prevent overfitting. By applying our chosen cross-validation strategy according to Section 2.5.2, each observation is out-of-sample in exactly one of the data folds. The observations in $\mathcal{D}_k$ are out-of-sample for the data fold k. In data fold k, the chosen model is trained on $\mathcal{D} \setminus \mathcal{D}_k$ and then used to predict the corresponding hold out test data $\mathcal{D}_k$. By following this approach, we get one claim frequency prediction per modeling technique for each observation in the whole dataset $\mathcal{D}$.

Further, we assume that we have two competing models, one called the benchmark model and the other the alternative model. These two models give us two predictions, we call these predictions p^{bench} and p^{alt} for the benchmark model and the alternative model respectively. Commonly, one uses predictions of the loss cost (the risk premium), but since we work with claim frequency in this thesis we will instead use predictions of the number of claims. We define the relativity r_i as the ratio of the prediction of the alternative model divided by the prediction of the benchmark model for observation i according to

$$r_i = \frac{p_i^{alt}}{p_i^{bench}}$$

Now we can explain the measures of model lift we will make use of in our study. These are called quantile plots and double lift charts.

2.5.4.1 Quantile plots

Quantile plots are a visual representation of the ability of a model to accurately differentiate between the best and the worst risks (Goldburd, Khare, & Tevet, 2016). Quantile plots, for comparison of claim frequency models, can be created by the following steps:

1. sort observations from smallest to largest relativity r_i
2. bin observations into groups of equal exposure e. Common choices for the number of bins are 5, 10 or 20 bins.
3. within each bin, calculate the ratio of the actual number of claims divided by the predicted number of claims with the benchmark model
4. for each quantile, plot the ratio in step 3

If the benchmark model is accurate, the ratio in the bins according to step 3 should be close to 100 %. However, if we can spot an upwards trend in the ratios, this would indicate that policies with a higher (lower) number of predicted claims in the alternative model also are those with higher (lower) proportion of actual claims in the benchmark model, which means that the alternative model better reflects the actual risk.

2.5.4.2 Double lift charts

Double lift charts are similar to quantile plots, but instead directly compares two models (Goldburd, Khare, & Tevet, 2016). Double lift charts, for comparison of claim frequency models, can be created by the following steps:

1. sort policies from smallest to largest relativity r_i
2. bin policies into groups of equal exposure e
3. within each bin, calculate the number of actual claims (c) and the number of predicted claims with both the benchmark model (p^{bench}) and the alternative model (p^{alt})
4. within each bin, calculate the percentage error for both models as $p/c - 1$
5. for each quantile, plot the percentage error in step 4

The best model is the model where the percentage error is closest to zero.

CHAPTER 3
METHOD

3.1 Simulated insurance claims count data

The study uses simulated home insurance claims data inspired by the characteristics found in a real data set provided by the insurance company Hedvig. The main advantage of this is that we will know the true underlying data generating function and measure errors exactly. To be able to study non-linear effects on the claim frequency, the simulated data incorporates two artificial two-way interactions. The simulation study has a "base case" simulation used for fitting, interpreting and comparing the models by using the tools and methods presented in Section 2.4 and 2.5. Further, the study includes variations of the base case simulation for scenario impact analysis. These different simulations will now be explained.

3.1.1 Base case simulation

The claim frequency $\boldsymbol{\lambda}(\boldsymbol{x})$ is simulated according to the following formula:

$$\begin{aligned}\log \boldsymbol{\lambda}(\boldsymbol{x}) = &-\frac{567}{120} + \frac{2}{5}x_{ageph} - \frac{6}{400}x_{ageph}^2 + \frac{1}{5000}x_{ageph}^3 - \frac{1}{10^6}x_{ageph}^4 + \frac{49}{200}x_{nbrcoi} \\ &+\frac{3}{2000}x_{size} - \frac{15}{100}x_{rental} - \frac{15}{100}x_{student} + 10\frac{x_{nbrcoi}}{x_{ageph}} + \frac{3}{400}x_{size}(1 - x_{rental})\end{aligned} \tag{3.1}$$

where the explanatory variables are defined in Table 3.1. The explanatory variables will often only be referred to by their index, e.g. *ageph* for x_{ageph}.

The explanatory variables were sampled from distributions inspired by the empirical distributions of the same variables in the real data.

The univariate frequencies in the model for each of the explanatory variables are illustrated in Figure 3.1 and 3.2.

Variable	Description	Unit	Domain
x_{ageph}	the age of the policyholder	years	$x_{ageph} \in [18, 100]$
x_{nbrcoi}	the number of people co-insured	count	$x_{nbrcoi} \in [0, 4]$
x_{size}	size of the apartment	square meters	$x_{size} \in [10, 280]$
x_{rental}	1 (or "Y") for rental apartments and 0 (or "N") for non-rental apartments	dummy	$x_{rental} \in [0, 1]$
$x_{student}$	1 (or "Y") for students and 0 (or "N") for non-students	dummy	$x_{student} \in [0, 1]$

Table 3.1: Summary of the explanatory variables in the simulated data

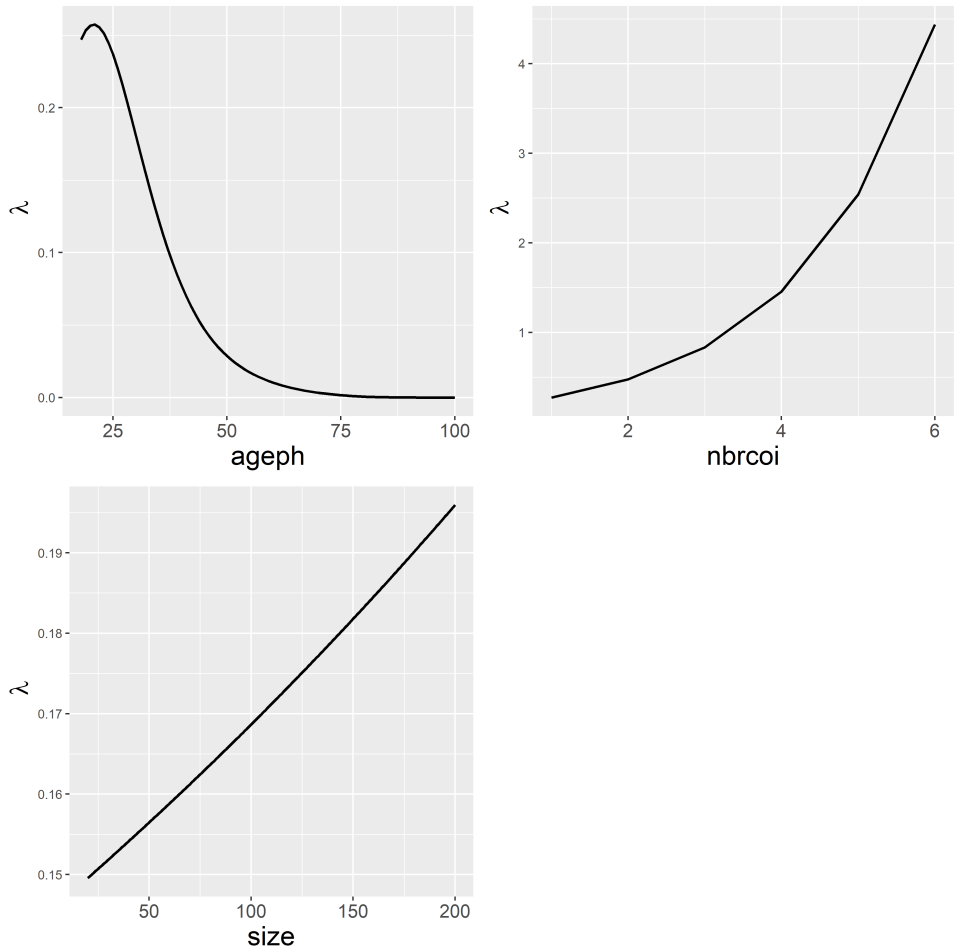

Figure 3.1: Frequencies of policyholder age (ageph), the number of people co-insured (nbrcoi) and apartment size (size)

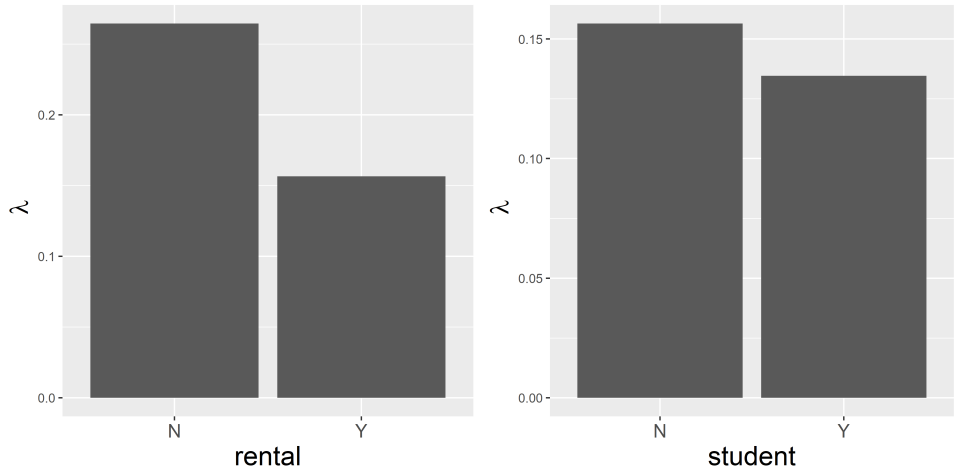

Figure 3.2: Frequencies of rental or non-rental apartments (rental) and students or non-students (student)

Based on the frequency function (3.1) we generate the observations $N_1, ..., N_n$ giving the data

$$\mathcal{D} = \{(N_1, \boldsymbol{x}_1, v_1), (N_2, \boldsymbol{x}_2, v_2), ..., (N_n, \boldsymbol{x}_n, v_n)\} \tag{3.2}$$

To be able to simulate from frequency function (3.1) we need to specify a claims count distribution. We choose the Poisson distribution, which means that the data $\mathcal{D}$ is simulated by independently generating observations according to

$$N_i \overset{\text{ind.}}{\sim} Poi(\lambda(\boldsymbol{x}_i)v_i) \text{ for } i = 1, ..., n \tag{3.3}$$

For simplicity we assume the exposures v_i are one for all i. In the base case simulation 50 000 observations were simulated according to Equation (3.3).

Figure 3.3 shows the distribution of claims in the simulated data of the base case.

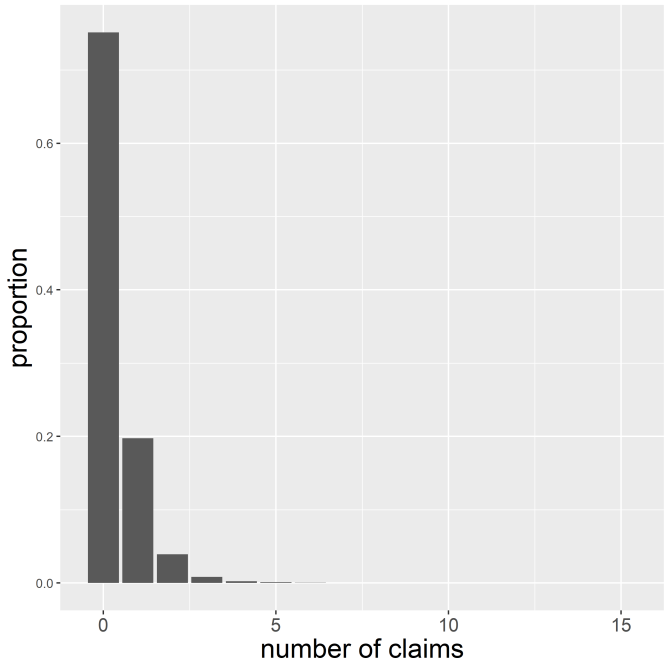

Figure 3.3: Distribution of the number of claims for simulated data

3.1.2 Variations of the base case simulation

Besides the base case simulation a number of alternative scenarios were tested in order to analyze the impact on the model performance results.

3.1.2.1 Alternative number of simulated observations

First, we wanted to investigate the impact of using different number of simulated observations. In this regard, we used the same frequency function given by Equation (3.1) and compared model performance with 2 500, 5 000, 10 000 and 25 000 simulated observations by Equation (3.3) with the base case (50 000 observations).

3.1.2.2 More complex interaction between policyholder age and number of co-insured

The base case frequency function given in Equation (3.1) have interaction effects with quite simple functional form. This is seldom the case in real data. In order to evaluate the ability of the GBM to find more complex interaction patterns, we redefined the interaction between policyholder age and the number of co-insured and then simulated

50 000 observations from the frequency function given in (3.4).

$$\begin{aligned}\log \boldsymbol{\lambda}(\boldsymbol{x}) = &-\frac{567}{120} + \frac{2}{5}x_{ageph} - \frac{6}{400}x_{ageph}^2 + \frac{1}{5000}x_{ageph}^3 - \frac{1}{10^6}x_{ageph}^4 \\ &+\frac{49}{200}x_{nbrcoi} + \frac{3}{2000}x_{size} - \frac{15}{100}x_{rental} - \frac{15}{100}x_{student} \\ &+\exp(-\frac{x_{ageph}}{10} + \sin^2(x_{nbrcoi})) + \frac{3}{400}x_{size}(1 - x_{rental})\end{aligned} \tag{3.4}$$

3.1.2.3 Negative-binomially distributed claims data

For simplicity we have assumed that the claims are Poisson distributed in the base case simulation. However, in real data the Poisson assumption of equal mean and variance seldom is completely fulfilled. Rather, so called over-dispersion is common which means that the variance is larger than the mean for the dependent variable. Therefore, we wanted to investigate the performance of the GBM given that claims are distributed more realistically. The negative-binomial distribution is able to account for over-dispersion through an extra parameter γ. To check for over-dispersion we estimated a quasi-Poisson model on the real claims data, which gave the dispersion parameter $\gamma = 1.4$. We then simulated 50 000 observations from the negative-binomial distribution given by Equation (3.5) which we used for fitting the models.

$$N_i \overset{\text{ind.}}{\sim} NegBin(\lambda(\boldsymbol{x}_i)v_i, 1.4) \text{ for } i = 1, ..., 50\,000 \tag{3.5}$$

3.2 Model training

For each simulated dataset $\mathcal{D}$ we perform the data partitioning and stratification by claim frequency as explained in Section 2.5.2 to get six equally sized subsets $\mathcal{D}_1, ..., \mathcal{D}_6$ with similar distribution of the claim frequency. This gives us six data folds, where $\mathcal{D} \setminus \mathcal{D}_k$ is used for model training and $\mathcal{D}_k$ is the hold-out test data for data fold k, $k \in \{1, ..., 6\}$. To fit the models, all of the explanatory variables as given in Table 3.1 were included. The details of the model training for the GLMs and the GBM models will now be explained.

3.2.1 Generalized linear models

Since we study claim frequency we followed the GLM assumptions according to Section 2.2.3 and 2.2.5. Thus, we use a log-link Poisson GLM for modelling the claim

frequency. For every data fold k three different GLMs were trained, referred to as GLM1, GLM2 and GLM3. GLM1 only includes the main effects of the true frequency functions; GLM2 also has the main effects but adds the interaction between *ageph* and *nbrcoi*; and GLM3 has both interactions except for the main effects. Thus, GLM3 would be very close to the true frequency functions. The main reason for including these three different GLMs was to get an understanding of the level of performance of the GBM. Our aim was to build benchmark pricing models that follow industry standards in order to make a fair comparison with the GBM. Since the frequency for policyholder age is non-linear, we used a binning approach of the policyholder age variable x_{age} before training. We did this by iteratively testing different binnings in order to find the best binning of the variable given the data for each simulation.

3.2.2 GBM model training

In order to fit the GBM models, nested stratified 6-fold cross-validation was applied according to Section 2.5.2, with detailed explanation in Algorithm 2. For every data fold we perform 5-fold cross-validation in order to choose the set of hyperparameters giving the lowest Poisson deviance loss. This will give us the particular GBM model chosen for each data fold.

3.2.2.1 Choice of hyperparameters

Since the performance of the GBM is significantly impacted by the choice of hyperparameters, a grid search strategy of the most important hyperparameters was applied for each of the data folds. The strategy used is traditional (or "cartesian") grid search, which means that a set of values for each hyperparameter is specified that we want to search over, and a model is trained for every combination of the hyperparameter values. In Algorithm 2 this iteration is defined by the foreach-loop. In a GBM it is important to find the optimal number of trees, M in Algorithm 1, since a higher number of trees for a particular learning problem can lead to overfitting. Another important parameter is the tree depth d, since it affects the maximum number of potential leaves in a tree. A larger value of d results in larger variable interaction effects. If d is too large we also increase the risk of overfitting the model.

For these reasons we apply grid search for M and d when fitting the GBMs. The following search grid was used to find the optimal number of trees M and tree depth d

for each data fold:

$$\begin{aligned} M &\in \{100, 200, ..., 2000\} \\ d &\in \{1, 2, 3, 4, 5\} \end{aligned} \tag{3.6}$$

This implies that for every data fold k we fit 100 models (20 levels of M times 5 levels of d). The learning rate, τ in Algorithm 2, does not have an optimal value since lower values always work better given that we train on sufficient number of trees. However, lower values of τ does increase the computation time. The learning rate should be set sufficiently low in order to get a good fit with reasonable computation time. Our choice was to set $\tau = 0.01$ since it fulfilled these criteria. Further, the subsampling parameter δ was set to 0.75 which means that for every tree fitted in the boosting procedure 75 % of the data will be randomly selected. This should reduce the risk of overfitting the models. We also set a threshold, κ, for the minimum number of observations in the tree nodes for a split to occur to reduce the risk of overfitting. We choose to set $\kappa = 0.01$ which implies that a split in a tree is not allowed if a resulting node would have less than 1 % of the observations in the training data.

3.3 Implementation

The methods were implemented in the statistical software R with the application of H2O. H2O is an open-source machine learning platform with an R-interface and incorporates algorithms for both GLM and GBM (as well as many other models and methods). H2O is available by installation of the package h2o in R. The main reasons for using H2O for this thesis are performance gains in computations and being able to do most of the modeling within the same package. For the most part we utilized H2O for model training. We also used H2O to compute variable importance and partial dependence. To compute Friedman's H-statistic we applied the iml (short for Interpretable Machine Learning) package. Some examples of R code from the implementation can be found in Appendix A.

CHAPTER 4

RESULTS

In this chapter the results of the simulation study are presented. The main focus will be on the base case simulation, from which we will present the results of the model training, the interpretation of the models and model performance comparisons. After this we will present the results of the alternative simulations.

4.1 Model training

In the following section we present the results of training the models in the base case simulation.

4.1.1 Generalized linear models

Table 4.1 shows the regression output of the model trained in data fold 4 of the base case simulation with only the main effects (GLM1), i.e. the model trained when data $\mathcal{D}_4$ was kept as test data. The binning of *ageph* is given by the new variable *agephc* seen in the output. From the output we can see that all model parameters are significant at $p = 0.05$. The regression output for the other five GLMs with the main effects of the base case simulation can be found in Appendix B.

name	coefficient	SE(σ)	z-score	p-value
Intercept	-1.01	0.03	-33.04	0.00
agephc.[20-24)	0.06	0.03	2.02	0.04
agephc.[24-28)	-0.07	0.03	-2.42	0.02
agephc.[28-31)	-0.28	0.03	-8.29	0.00
agephc.[31-34)	-0.54	0.04	-14.59	0.00
agephc.[34-37)	-0.87	0.04	-20.46	0.00
agephc.[37-41)	-1.14	0.05	-24.82	0.00
agephc.[41-45)	-1.60	0.06	-25.17	0.00
agephc.[45-50)	-2.13	0.10	-22.35	0.00
agephc.[50-55)	-2.65	0.17	-15.61	0.00
agephc.[55-58)	-2.36	0.29	-8.15	0.00
agephc.[58-100)	-3.84	0.71	-5.44	0.00
rental.Y	-0.63	0.02	-36.24	0.00
student.Y	-0.14	0.02	-7.48	0.00
nbrcoi	0.58	0.01	55.48	0.00
size	0.01	0.00	15.53	0.00

Table 4.1: Regression output from GLM1 fitted in data fold 4 of the base case simulation

The case also including the interaction effect between *ageph* and *nbrcoi* (GLM2) for data fold 4 is seen in Table 4.2. We see that the added interaction term $nbrcoi : ageph$ is significant at $p = 0.05$.

name	coefficient	SE(σ)	z-score	p-value
Intercept	-0.74	0.14	-5.20	0.00
agephc[20-24)	0.16	0.05	3.46	0.00
agephc[24-28)	0.13	0.07	1.81	0.07
agephc[28-31)	0.00	0.10	0.03	0.97
agephc[31-34)	-0.14	0.12	-1.13	0.26
agephc[34-37)	-0.30	0.15	-2.04	0.04
agephc[37-41)	-0.48	0.17	-2.74	0.01
agephc[41-45)	-0.70	0.21	-3.32	0.00
agephc[45-50)	-1.06	0.25	-4.19	0.00
agephc[50-55)	-1.35	0.33	-4.14	0.00
agephc[55-58)	-1.14	0.46	-2.51	0.01
agephc[58-100)	-2.38	0.79	-3.01	0.00
nbrcoi	0.96	0.04	22.84	0.00
size	0.00	0.00	15.10	0.00
rentalY	-0.60	0.02	-34.63	0.00
studentY	-0.15	0.02	-7.63	0.00
ageph	-0.02	0.01	-2.55	0.01
nbrcoi:ageph	-0.01	0.00	-9.36	0.00

Table 4.2: Regression output from GLM2 fitted in data fold 4 of the base case simulation

We also show the case when including both interactions from the true frequency function (GLM3) of the base case simulation for data fold 4. This is seen in Table 4.3. We see that both interaction terms $nbrcoi : ageph$ are highly significant at $p = 0.05$.

name	coefficient	SE(σ)	z-score	p-value
Intercept	-0.96	0.14	-6.76	0.00
agephc[20-24)	0.17	0.05	3.63	0.00
agephc[24-28)	0.14	0.07	1.97	0.05
agephc[28-31)	0.02	0.10	0.22	0.83
agephc[31-34)	-0.12	0.12	-0.95	0.34
agephc[34-37)	-0.28	0.15	-1.87	0.06
agephc[37-41)	-0.44	0.17	-2.54	0.01
agephc[41-45)	-0.66	0.21	-3.14	0.00
agephc[45-50)	-1.04	0.25	-4.09	0.00
agephc[50-55)	-1.32	0.33	-4.06	0.00
agephc[55-58)	-1.08	0.46	-2.38	0.02
agephc[58-100)	-2.31	0.79	-2.91	0.00
nbrcoi	0.95	0.04	22.91	0.00
size	0.01	0.00	21.65	0.00
rentalY	-0.15	0.04	-4.30	0.00
studentY	-0.14	0.02	-7.47	0.00
ageph	-0.02	0.01	-2.71	0.01
nbrcoi:ageph	-0.01	0.00	-9.34	0.00
size:rentalY	-0.01	0.00	-14.62	0.00

Table 4.3: Regression output from GLM3 fitted in data fold 4 of the base case simulation

4.1.2 Cross-validation of GBM models

The results for the 10 best performing models, in terms of the lowest residual Poisson deviance, of the 5-fold cross-validation for data fold 4 can be seen in Table 4.4. The best model for this data fold has 1700 trees fitted to the data with maximum tree depth of 2. This means that at the highest two-way interactions are allowed for in each fitted tree. Since we only have two-way interactions in our frequency function from Equation (3.1) this is the correct selection of the tree depth. The same results for the other five data folds can be seen in Appendix C. From these results we see that for half of the data folds models with maximum tree depth of 3 give the lowest residual deviance. Hence, in these cases the selected models will allow for three-way interactions which we do not have in the true frequency function. This indicates some overfitting done by the GBM.

number of trees (M)	tree depth (d)	model id	residual deviance
1700	2	fold_4_model_82	1.188917
1800	2	fold_4_model_87	1.188917
1900	2	fold_4_model_92	1.188917
2000	2	fold_4_model_97	1.188917
1600	2	fold_4_model_77	1.188923
1500	2	fold_4_model_72	1.188941
1400	2	fold_4_model_67	1.188996
1400	3	fold_4_model_68	1.189038
1900	3	fold_4_model_93	1.189048
2000	3	fold_4_model_98	1.189048

Table 4.4: Grid search cross-validation results of the 10 best performing models for data fold 4 of the base case simulation. For all models the learn rate (τ) = 0.01, sample rate (δ) = 0.75 and min observations (κ) = 0.01.

4.2 Model interpretation

In this section we will focus on model interpretation of the GBM. To get a better understanding for which variables the GBM regards as most important for estimating the frequency we will use the variable importance measure. After this, we will look at partial dependence plots and H-statistics to gain insight into potential effects for the claim frequency and the ability of the GBM to identify the interactions in the data.

4.2.1 Variable importance

Figure 4.1 shows the variable importance over all six data folds with the GBM model and the GLM respectively. The variables have been ranked from top to bottom according to the average variable importance over the folds. Variables with zero importance are ordered alphabetically from Z to A.

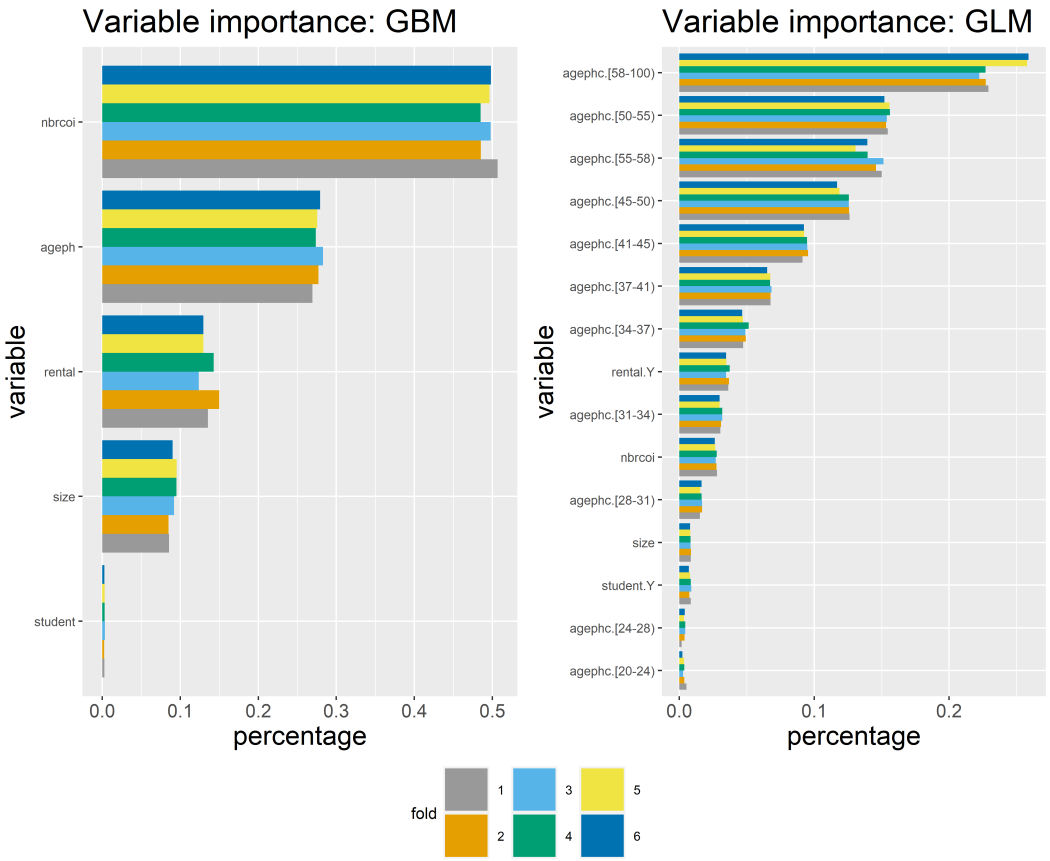

Figure 4.1: Variable importance

The variable importance of the GBM and the GLM are not directly comparable since the models work differently. However, it indicates that the GBM puts larger weight especially on the number of people co-insured for predicting the claim frequency. The age of the policyholder are identified by both models to be important for predicting the claim frequency. Both the GBM and the GLM considers the student variable to be relatively unimportant.

4.2.2 Partial dependence

Next, we analyze the effect of the explanatory variables on the claim frequency by partial dependence plots for both the GLM and the GBM for each of the data folds.

The partial dependence for policyholder age and the number of people co-insured is displayed in Figure 4.2. The two top panels show the partial dependence for the age of the policyholder and the bottom panels for the number of people co-insured.

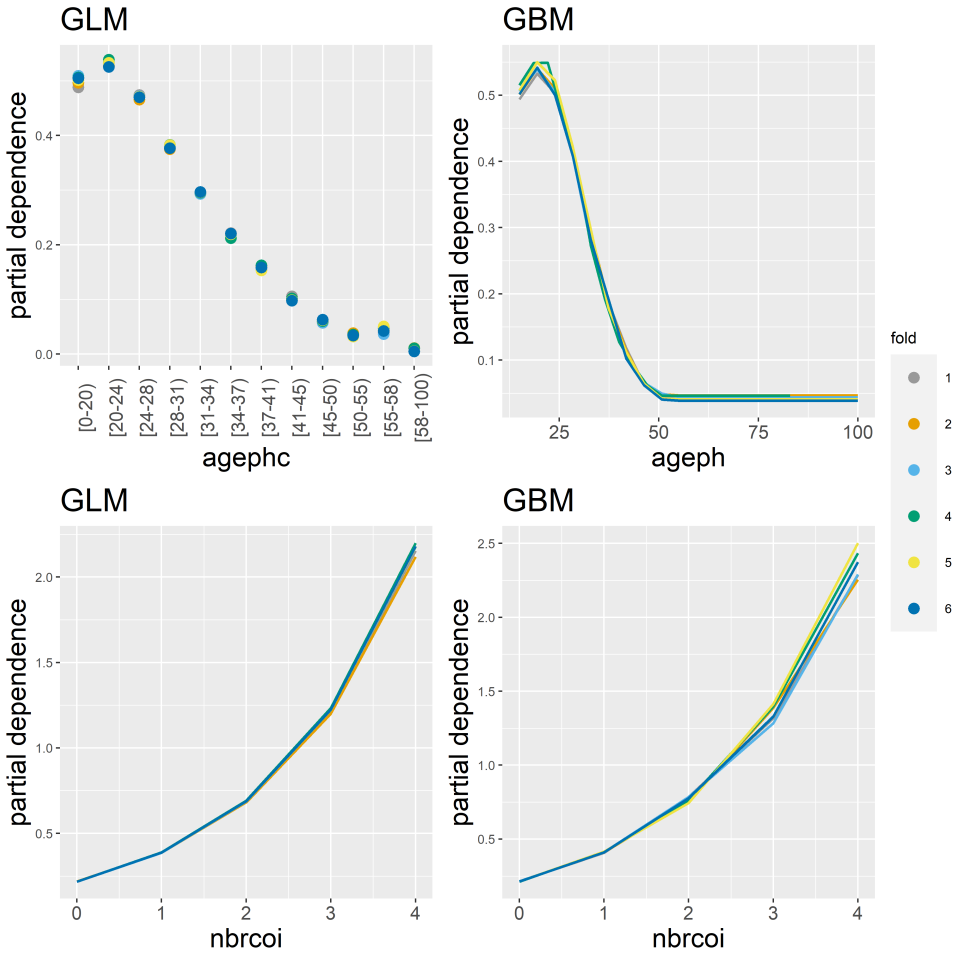

Figure 4.2: Partial dependence for agephc / ageph and nbrcoi

As we expect from the definition of the age categories in the GLM we see that the partial dependence of the claim frequency increase slightly between the youngest age categories and then decrease after that. We see in the top right panel that the GBM has found a similar effect of the the policyholder age on the claim frequency. For the number of people co-insured we also note that the GLM and the GBM are showing similar patterns, i.e. that the claim frequency increase with the number of people co-insured. We can also see from the plots of the number of people co-insured that the effect of the GBM is less stable across the folds compared to the GLM. This indicates some overfitting tendencies of the GBM.

Figure 4.3 show the partial dependence plots for the variables apartment size and rental. We note that both models also in this case show similar effect patterns. For apartment size the claim frequency increase with increasing apartment size. However, also for apartment size the GBM is less stable across the folds compared to the GLM. For

the GBM, we see that the claim frequency increase up to about 150 square meters and then it levels off. In the case of the GLM the claim frequency instead continue to increase since the effect is defined as continuous in the model. Hence, the GBM could be a bit more robust compared to the GLM with regard to extrapolation of the effects.

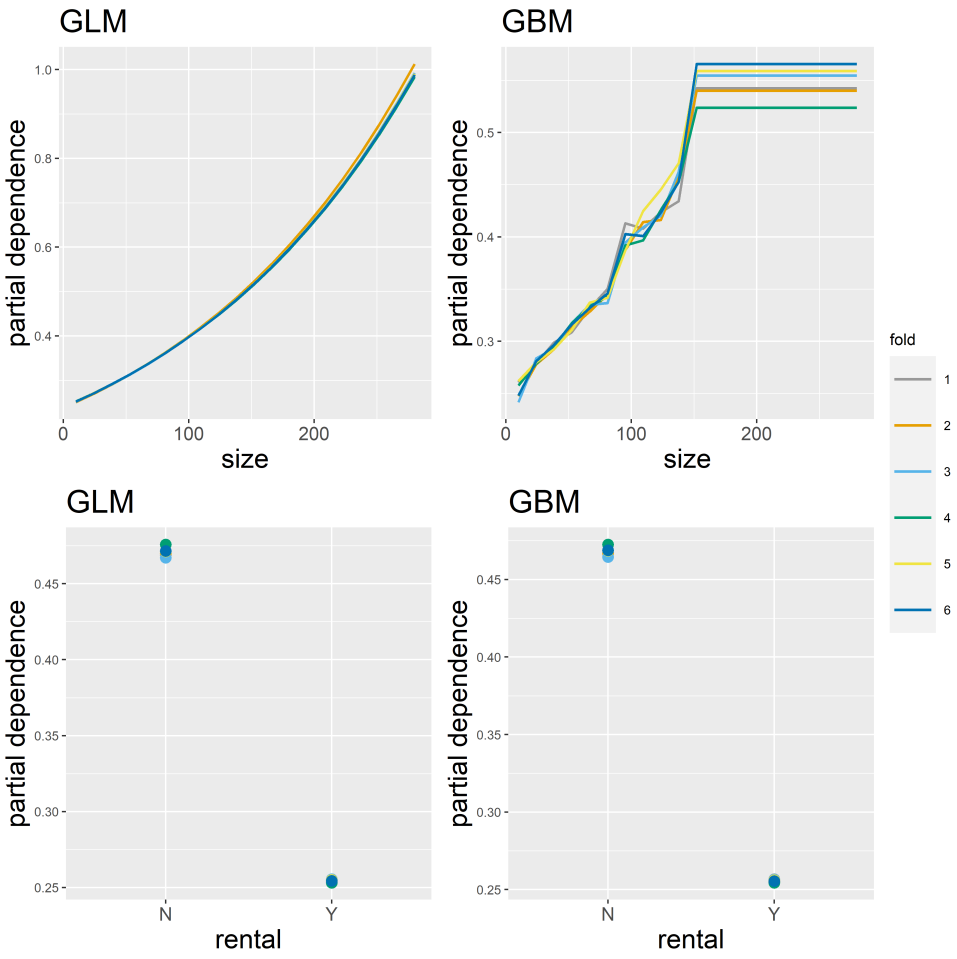

Figure 4.3: Partial dependence for size and rental

In Figure 4.4 the partial dependence for the student effect for the models can be seen. The effect is modeled similarly for the two models where the effect of being a student (=Y) results in lower predicted claim frequency compared to non-students (=N). The partial dependence of being a student (=Y) is slightly higher in the GBM compared to the GLM.

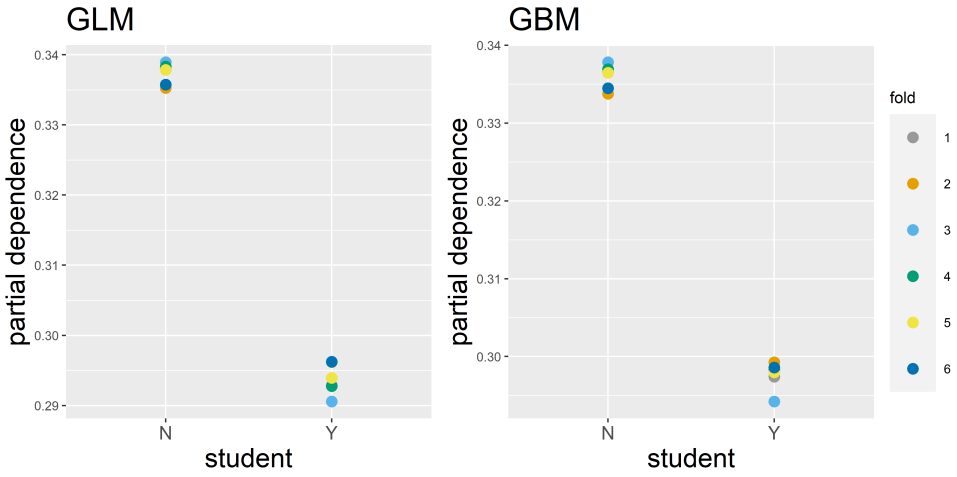

Figure 4.4: Partial dependence for student

4.2.3 Finding interactions

The interaction strength from the variables were computed with Friedman's H-statistic based on the GBM. Figure 4.5 shows the overall interaction strength across data folds for each of the explanatory variables. The results show that *ageph*, *nbrcoi*, *rental* and *size* are each heavily involved in interactions with other variables. The age of the policyholder and number of people co-insured give the strongest interaction signals. For *ageph* more than 60 % of the prediction variance stems from interaction effects with other variables. Since these variables are all involved in interactions according to the true frequency function, given in Equation (3.1), this is what we would expect. The student variable, which is not actually involved in any interaction, gives no evidence of interaction effects involving other variables.

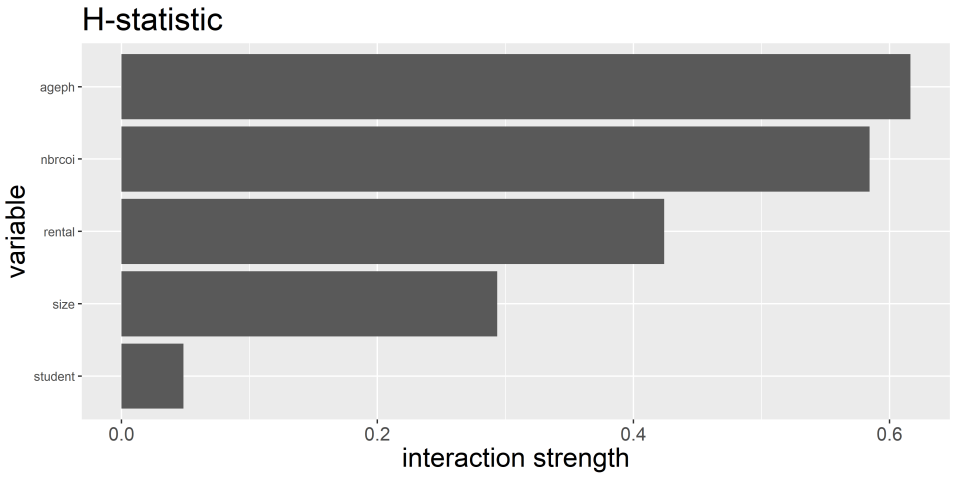

Figure 4.5: Overall interaction strength of each explanatory variable by calculation of the H-statistic

Since we now have evidence of strong interaction signals in the data it is of interest to determine the particular other variables with which each one interacts. In order to identify any two-way interactions we compute the two-way H-statistic for all explanatory variables. This gives 10 combinations of different variable pairs. The two-way interaction signals are ranked from strong to weaker in Table 4.5.

Table 4.5: Ranking of two-way interaction signals

Variables	H-statistic
rental:size	0.51
nbrcoi:ageph	0.50
rental:nbrcoi	0.26
rental:ageph	0.24
size:ageph	0.12
size:nbrcoi	0.08
student:nbrcoi	0.06
student:ageph	0.06
student:size	0.03
student:rental	0.01

The combination of the variables policyholder age and number of co-insured together with the combination of size and rental give by far the strongest two-way interaction signals according to the H-statistic. This is reassuring, since these are the actual two-way interactions included in our true frequency function. However, we do see that rental and number of co-insured and rental and policyholder age also indicate relatively strong two-way interactions even though they are not included as interactions in the true frequency function.

To analyze the interaction of these four variable pairs further we can again look at partial dependencies. We now plot the partial dependence of the first variable grouped by values of the second variable for these four variable pairs. Interaction effects between the variable pairs can be discovered by comparing the pattern of the partial dependence curves over the different groups. The plots provide evidence of a potential interaction when the pattern is significantly different for policyholders in different groups.

Figure 4.6 shows the partial dependence of the number of co-insured grouped by policyholder age in the left upper panel and the right upper panel displays the same information but also including the one standard error for the partial dependence. The two bottom panels displays the same information for the partial dependence of apartment size grouped by rental.

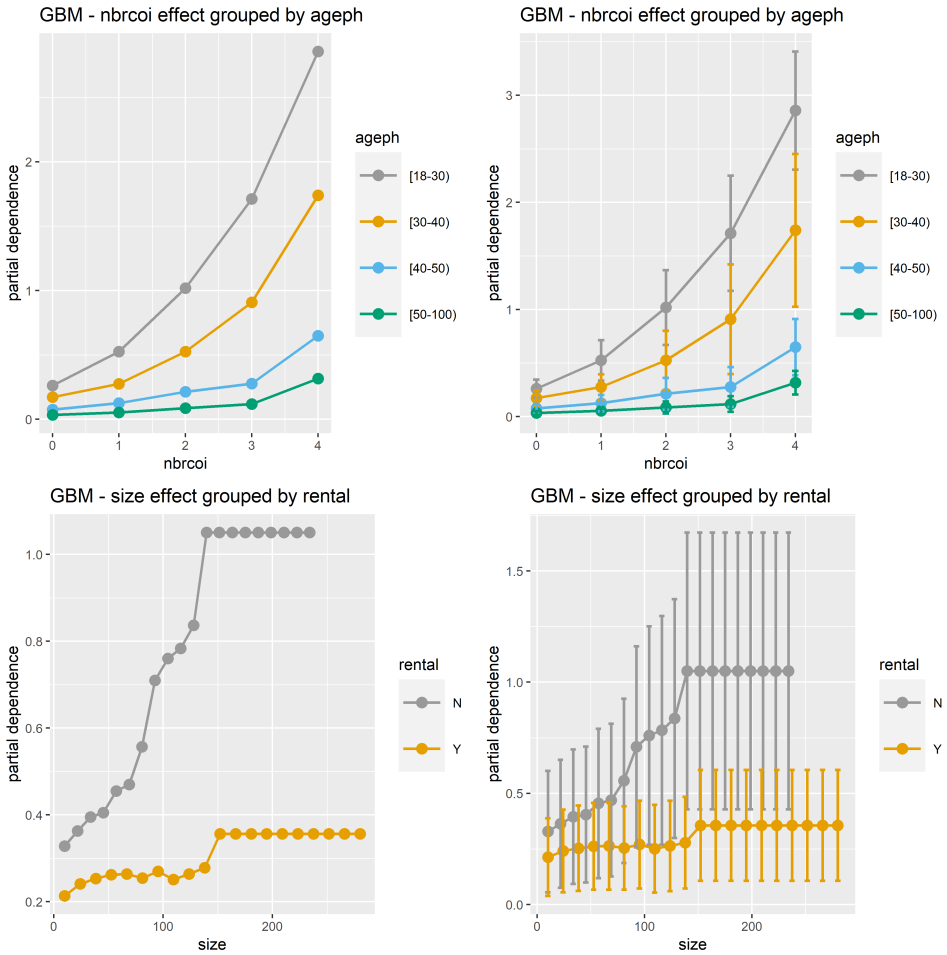

Figure 4.6: Partial dependence of the number of co-insured grouped by a the age of policyholders

We see that the claim frequency increases more over the number of co-insured in the lower policyholder age categories compared to higher ages. It is also evident that the claim frequency increases more over apartment size for non-rentals (=N) compared to rentals (=Y). From the right plots, which show the mean effects including the one standard error bars, we see few cases where the intervals overlap the mean effect for other groups. This indicates a significant differentiation in risk for both of these variable pairs by the GBM.

The partial dependence of policyholder age grouped by rental and the partial dependence of number of co-insured grouped by rental can be seen in 4.7.

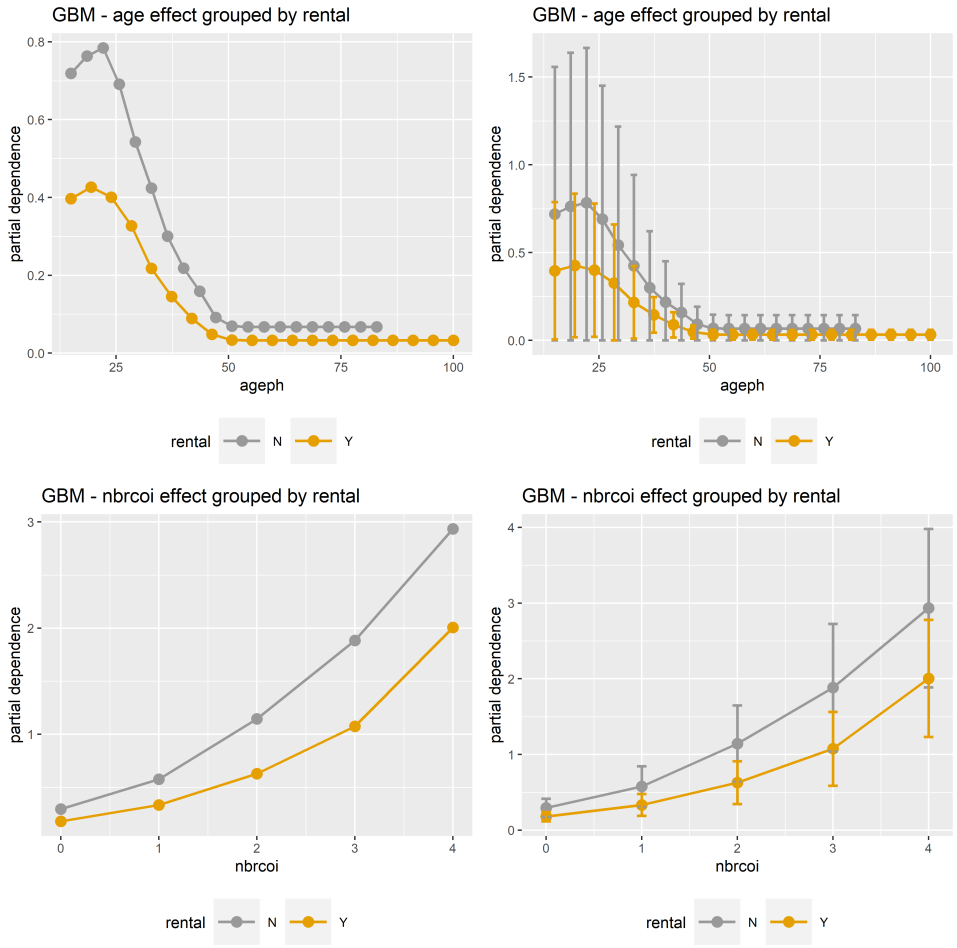

Figure 4.7: Partial dependence of the age effect grouped by the number of co-insured

We do see a bit different patterns for these variable pairs when only looking at the two left panels. However, for most data points the one standard error intervals overlap the mean effect for the other group which indicates small or no significant differentiation in claim frequency by the GBM. Hence, the analysis provide strong evidence of two-way interactions between ageph and nbrcoi and between rental and size. Some support for the interaction between ageph and rental and between nbrcoi and rental is also found by the H-statistics, but partial dependencies gives no evidence of any significant interactions at play for these variable pairs.

4.3 Model performance

4.3.1 Generalization loss

For assessing the statistical performance of the models the out-of-sample Poisson deviance for each of the six data folds was compared for the true model, the null model,

the GLM without any interactions (GLM1), the GLM including the correctly "discovered" interaction of policyholder age and number of people co-insured (GLM2), the GLM including both correct interactions (GLM3) and the GBM. The true model corresponds to the true expected frequencies, which we know since we work with simulated data. By knowing the true model we have the great advantage that we can explicitly quantify the quality of all estimated models. The null model corresponds to a model which only includes an intercept term, thus predicting the overall portfolio average for all observations. Table 4.6 summarises these results.

Table 4.6: Out-of-sample Poisson deviance

	fold						
	1	2	3	4	5	6	avg
true	0.7659	0.7611	0.7795	0.7710	0.7692	0.7722	0.7698
null	0.9542	0.9613	0.9639	0.9518	0.9486	0.9547	0.9557
glm1	0.7743	0.7718	0.7892	0.7774	0.7794	0.7793	0.7786
glm2	0.7719	0.7694	0.7860	0.7752	0.7776	0.7777	0.7763
glm3	0.7679	0.7638	0.7814	0.7720	0.7711	0.7721	0.7714
gbm	0.7675	0.7666	0.7840	0.7736	0.7719	0.7759	0.7733

We see that all GLMs and the GBM outperform the null model since they all have lower deviance than the null model. Since GLM2 and GLM3 have lower deviance than GLM1, it is evident that adding the interactions to the model resulted in a better fit. Accept for the true model, we also see that the GBM and GLM3 have the lowest deviance of all models, thus giving the highest prediction accuracy. GLM3 performs better than the GBM. This is not surprising since GLM3 is very close to the actual data generating function and in addition that the GBM seems to slightly overfit the data.

In order to more clearly see the differences between the models, Figure 4.8 shows the improvements in mean out-of-sample deviance compared to the null model for each of the data folds. It is evident that the GBM consistently performs better than GLM1 and GLM2 for every data fold, but slightly worse than GLM3. We can also see that there is still room for improvement when comparing the models to the true model.

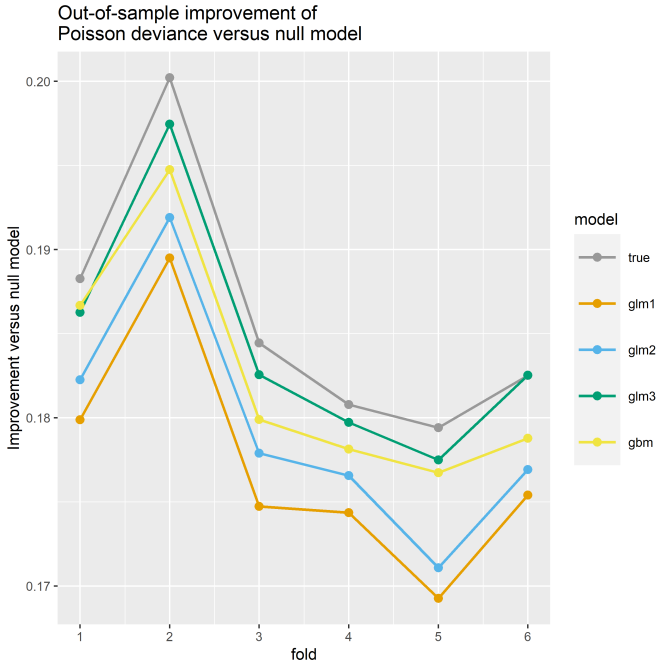

Figure 4.8: Improvements of the mean sample deviance versus the null model

Since we know the true model and the true expected frequency function we can compute the estimation loss with respect to the true model. This allows us to assess if we draw the right conclusions based on the cross-validation analysis. Table 4.7 shows the out-of-sample Poisson estimation loss for all estimated models.

Table 4.7: Out-of-sample Poisson estimation loss

	fold						
	1	2	3	4	5	6	avg
null	0.1896	0.1829	0.1851	0.2131	0.1819	0.1876	0.1901
glm1	0.0090	0.0083	0.0087	0.0101	0.0084	0.0080	0.0087
glm2	0.0065	0.0063	0.0062	0.0067	0.0063	0.0062	0.0064
glm3	0.0011	0.0012	0.0011	0.0014	0.0012	0.0012	0.0012
gbm	0.0043	0.0037	0.0048	0.0046	0.0029	0.0043	0.0041

The results align well with the results of analysing the Poisson deviance. GLM3 gives the lowest estimation loss. The GBM performs well and is close to GLM3 in terms of prediction accuracy. Both GLM2 and GLM3 have lower estimation loss than GLM1, thus confirming that adding the interactions leads to better prediction accuracy.

4.3.2 Model lift

We also want to evaluate the potential economic value of applying the models for creating pricing tariffs. As suggested in Section 2.5.4 this can be done by analyzing model

lift. Here, model lift refers to the ability of a model to prevent adverse selection.

Figure 4.9 shows quantile plots of model lift for comparing the three GLMs with the GBM. The policies have been sorted according to increasing relativity of the predicted frequency of the alternative model divided by the predicted frequency of the benchmark model and then binned into five groups of equal exposure size (indicated by the grey bars plotted on the left y-axis). The x-axis displays the span of these relativities in each of the bins. The blue line, which is plotted against the right y-axis, is the proportion of actual claims to the predicted number of claims under the benchmark model.

The two top panels compares GBM with GLM1, where the top left panel has GLM1 as the benchmark model and the top right panel the GBM as the benchmark model. The top right panel shows rather flat ratios, but the top left panel shows an increasing trend in the ratios. This increasing trend in the ratios implies that policies for which we would predict fewer claims under the alternative model (i.e. the GBM) compared to the benchmark model, the policies in the first bins, are policies with lower proportion of actual claims to predicted number of claims in the benchmark model. Further, policies having a higher number of predicted claims under the alternative model compared to the benchmark model, the policies in the last bins, also have high proportion of actual claims to predicted number of claims in the benchmark model. For these reasons, the GBM is able to spot deficiencies in the GLM. Since we have seen the GBM correctly has identified the interactions present in the true frequency function this would give the GBM an upper edge compared to GLM1. It is especially evident that the GBM is better than GLM1 when comparing the first and last bins.

The two middle panels compares GBM with GLM2. The trend in the ratio of actual claims to the predicted number of claims under GLM2, seen in the middle left panel, is now slightly flatter compared to the top left panel. The opposite holds when comparing the middle right panel with the top right panel. Hence, the GBM performs better than GLM2 as well, but it also implies that GLM2 performs better than GLM1 in ranking claim frequency risk. This would be expected since the GBM effectively includes an estimation of both of the actual interactions included in the true frequency function and GLM2 includes only one of them.

Finally, when comparing the GBM with GLM3, in the two bottom panels, we now see that the tables are turned. The trend in the ratio of actual claims to the predicted number of claims under GLM3, seen in the bottom left panel, is now even flatter compared to the top left panel and middle left panel. The bottom right panel shows a quite clear increasing trend in the proportion of actual claims to the predicted number of claims

under the GBM. Hence, GLM3 performs better than the GBM. Again, this would be expected since GLM3 almost replicates the true frequency function.

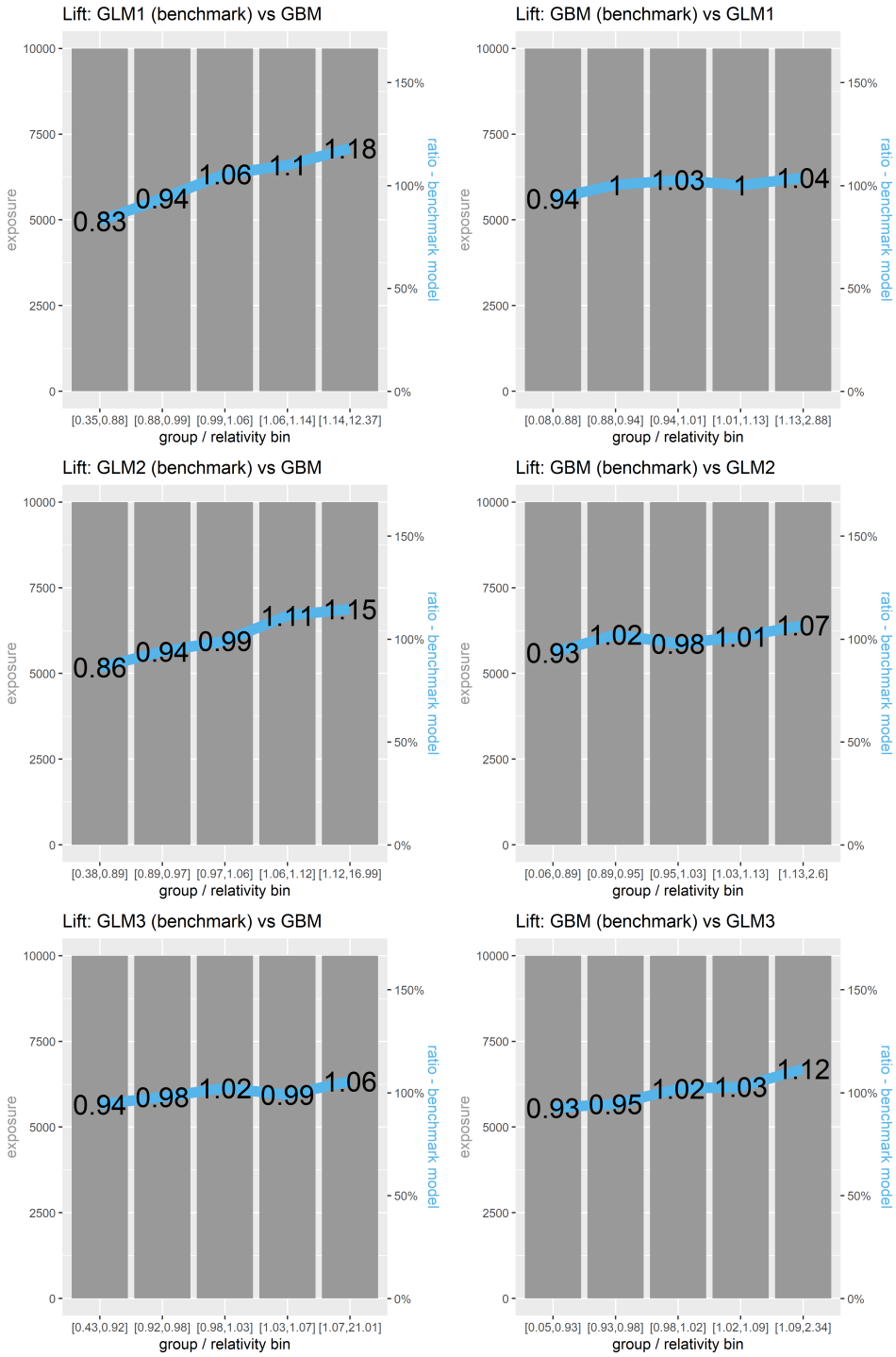

Figure 4.9: Assessment of model lift with quantile plots

These findings are supported by the double lift charts in Figure 4.10. The GBM has

an overall smaller percentage error over the relativity bins compared to both GLM1 and GLM2 (the two top panels), but performs slightly worse than GLM3 (the bottom panel). Since the percentage error of GLM2, in the top right panel, is lower than the percentage error for GLM1 in the top left panel GLM2 has better ranking ability than GLM1.

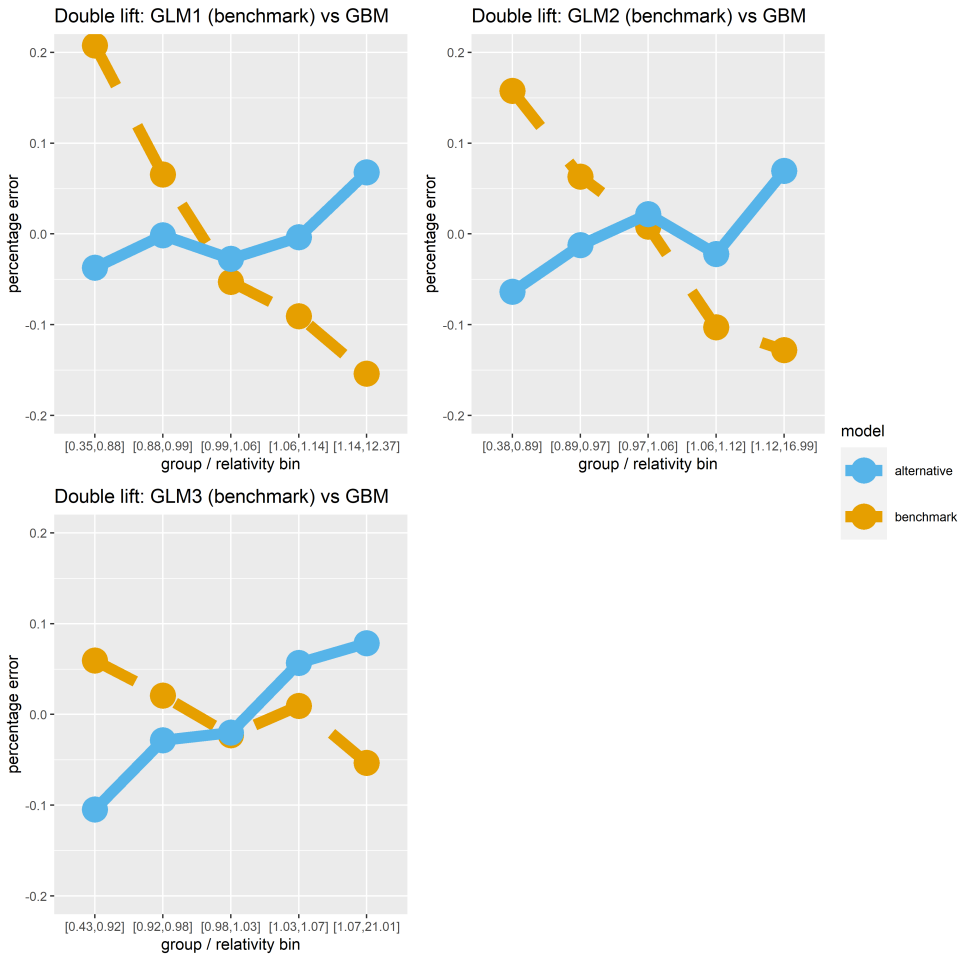

Figure 4.10: Assessment of model lift with double lift charts

4.4 Alternative scenarios

In this section we present the simulation results of the variations of the base case scenario.

4.4.1 Alternative number of simulated observations

We start by evaluating the impact of using alternative numbers of simulated observations. Table 4.8 shows the mean out-of-sample Poisson deviance over the data folds for 2 500, 5 000, 10 000, 25 000 and 50 000 observations. For up to 10 000 observations we

see that the GBM has higher Poisson deviance than GLM1, but with higher number of observations we see that the GBM starts to improve.

Table 4.8: Out-of-sample Poisson deviance

	number of simulated observations				
	2 500	5 000	10 000	25 000	50 000
true	0.7846	0.7626	0.7608	0.7681	0.7698
null	0.9636	0.9661	0.9610	0.9545	0.9557
glm1	0.7922	0.7782	0.7723	0.7760	0.7786
glm2	0.7926	0.7770	0.7716	0.7742	0.7763
glm3	0.7889	0.7670	0.7663	0.7695	0.7714
gbm	0.8035	0.7839	0.7736	0.7736	0.7733

Further, we analyze the mean out-of-sample Poisson estimation loss, seen in Table 4.9. It is evident that the estimation loss improves significantly for the GBM with an increasing number of observations. For 2 500 and 5 000 observations the GBM performs even worse than GLM1, likely due to overfitting. For 10 000 observations we can see that the GBM at least improves over GLM1 and for 25 000 and 50 000 observations it lies between GLM2 and GLM3 in terms of estimation loss.

Table 4.9: Out-of-sample Poisson estimation loss

	number of simulated observations				
	2 500	5 000	10 000	25 000	50 000
null	0.1722	0.1909	0.1945	0.1837	0.1901
glm1	0.0152	0.0133	0.0114	0.0104	0.0087
glm2	0.0178	0.0130	0.0093	0.0094	0.0064
glm3	0.0139	0.0086	0.0036	0.0042	0.0012
gbm	0.0205	0.0158	0.0099	0.0073	0.0041

4.4.2 More complex interaction between policyholder age and number of co-insured

We now investigate the case when we have have adjusted the interaction between policyholder age and number of co-insured according to Equation (3.4). The results of deviance and estimation loss are given in Table 4.10 and 4.11, comparing the case of the adjusted interaction with the base case. We can see that GLM2 has trouble taking this interaction into account since the deviance loss and estimation loss are not much different from GLM1. The GBM still seems to perform well, performing better than GLM2 but not quite as good as GLM3.

Table 4.10: Out-of-sample Poisson deviance

	Interaction type	
	Adjusted interaction	Base case
true	0.7474	0.7698
null	0.8318	0.9557
glm1	0.7528	0.7786
glm2	0.7526	0.7763
glm3	0.7487	0.7714
gbm	0.7496	0.7733

Table 4.11: Out-of-sample Poisson deviance

	Interaction type	
	Adjusted interaction	Base case
null	0.0834	0.1901
glm1	0.0054	0.0087
glm2	0.0051	0.0064
glm3	0.0014	0.0012
gbm	0.0020	0.0041

To check the two-way interaction strengths we compute the two-way H-statistic for all explanatory variables, seen in Table 4.12. Once again, the strongest two-way interactions are the actual interactions given by (3.4), i.e. the variable pairs *rental* : *size* and *nbrcoi* : *ageph*. Compared to the two-way H-statistics of the base case, seen in Table 4.5, *nbrcoi* : *ageph* has a lower H-statistic than *rental* : *size*. The reasons for this could be explained by the increased function complexity of the interaction, making it harder for the GBM to find relevant cut points for the tree nodes, but also because the interaction is now overall weaker compared to the base case. However, similar to the base case, *rental* : *nbrcoi* and *rental* : *ageph* also have relatively high interaction strengths according to the H-statistic even though they are not included as interactions in the true frequency function.

Table 4.12: Ranking of two-way interaction signals

Variables	H-statistic
rental:size	0.55
nbrcoi:ageph	0.20
rental:nbrcoi	0.18
rental:ageph	0.17
student:size	0.09
size:ageph	0.08
student:ageph	0.06
student:nbrcoi	0.05
size:nbrcoi	0.05
student:rental	0.05

We again check the grouped partial dependence curves for these four variable pairs, seen in Figure 4.11. In the top left panel, we see the partial dependence of *nbrcoi* grouped by categories of *ageph*. Similar to the base case, we see that the one standard error partial dependency intervals for different age groups do not overlap the mean effect for other groups, thus providing evidence of a significant interaction. We also see in the top right panel that *rental* : *size* still gives strong evidence of an interaction effect and from the bottom panels that *rental* : *nbrcoi* and *rental* : *ageph* still indicates small or no significant interaction.

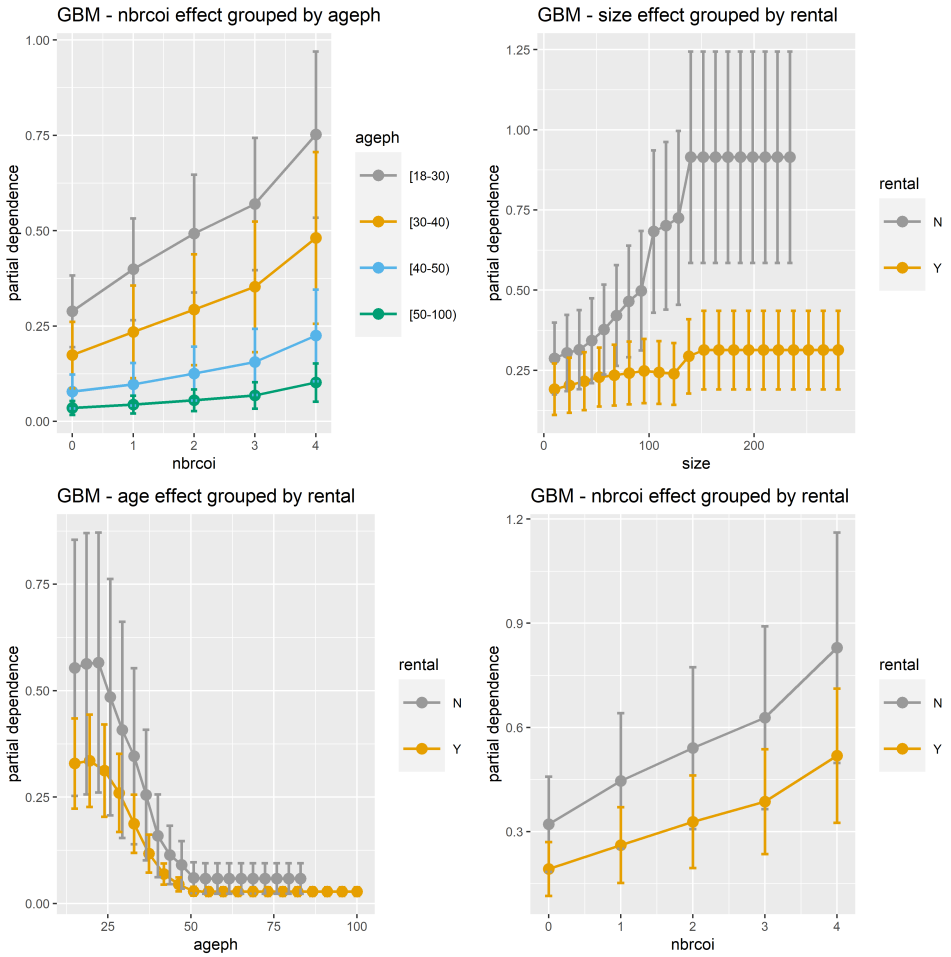

Figure 4.11: Partial dependence of of highest ranking two-way interactions

To summarise, the GBM still provides strong evidence of an interaction between *nbrcoi* and *ageph* given the frequency function defined by Equation (3.4). Even though we have not investigated it in detail, it seems the GLM does not work well for modeling the modified interaction of *nbrcoi* and *ageph*.

4.4.3 Negative-binomially distributed claims data

Next, we analyze the impact of changing the claims distribution from Poisson to the negative binomial distribution, as given by Equation (3.5). The results of the mean out-of-sample Poisson deviance and estimation loss are given in Table 4.13 and 4.14 respectively. It is evident that the deviance and estimation loss are higher for all models when the data is simulated from the negative binomial distribution. This is expected since the negative binomial distribution introduces more variance. Thus, except for overall higher deviance and estimation losses the scenario is similar to the base case when the claims are negative-binomially distributed.

Table 4.13: Out-of-sample Poisson deviance

	Claims distribution	
	Negative binomial	Poisson (base case)
true	0.8665	0.7698
null	1.0576	0.9557
glm1	0.8766	0.7786
glm2	0.8735	0.7763
glm3	0.8682	0.7714
gbm	0.8705	0.7733

Table 4.14: Out-of-sample Poisson estimation loss

	Claims distribution	
	Negative binomial	Poisson (base case)
null	0.1901	0.1901
glm1	0.0094	0.0087
glm2	0.0072	0.0064
glm3	0.0021	0.0012
gbm	0.0051	0.0041

CHAPTER 5
DISCUSSION

5.1 Objectives, revisited

A main objective of this thesis was to investigate and compare GBM models with traditional GLMs for modeling of the claim frequency. From the results of our simulation study, it was clear the selection of hyperparameters has a major impact on the performance of the GBM. Thus, it becomes important to carefully consider which particular hyperparameters to select for a given model. In this regard, we have shown that the application of cross-validation techniques and grid search strategies are helpful methods to find the best set of hyperparameters for a particular GBM model. However, this comes at the cost of increased computation times for training the models. This of course depends on the amount of available computation power, the selected grid search strategy, model complexity as well as the size of the training data. In our case, using a single personal computer, the time required to train the GBM models was measured in hours while the GLMs took seconds or minutes to train.

In terms of comparison of model performance from a statistical perspective, the analysis of the mean out-of-sample Poisson deviance and estimation loss showed that the GBM provides a significant boost in prediction accuracy compared to the GLM with only main effects in the base case simulation, as seen in Table 4.6 and 4.7. From the perspective of creating pricing tariffs, the model lift tests in Section 4.3.2 implies that the GBM has strong capability to rank risk and prevent adverse selection in a non-life insurance pricing setting, outperforming the GLM with only main effects. Thus, if we never tried to find out about these interactions and then included them in the GLM, the GBM has a strong upper hand in terms of both prediction accuracy and ranking of the claim frequency risk compared to the GLM. We also saw that the GBM performed better than the GLM including one of the interactions (GLM2) from the true frequency function. The GBM did not beat the GLM including both interactions (GLM3) from the

true frequency function, but this is not very surprising since this GLM lies very close to the true frequency function. Further, we discovered some overfitting being done by the GBM in terms of some of the main effects, such as apartment size (see Figure 4.3), and also some interaction effects according to Table 4.5.

The objective of this thesis was also to investigate available tools for model interpretation and to assess the interpretability of the results of gradient boosting in a non-life insurance pricing setting. In Section 4.2.1, we saw that by using a measure of variable importance we gain insights into what variables the GBM regards as most important for prediction of the claim frequency, similar to the parameter weights of the Poisson GLM. Further, by application of partial dependence plots, we got insights into how the variables in the GBM model affect the predicted claim frequency. These results were also in-line with the partial dependence plots of the GLM and what was expected given the true frequency function of the base case simulation. From this we saw the ability of the GBM to automatically find non-linear effects such as the effect of policyholder age on the claim frequency in Figure 4.2. For the GLMs, such non-linear continuous variables need some preparatory work before being correctly included in a GLM, e.g. by application of variable binning techniques and smoothing techniques. However, we did notice that the GBM was more likely to overfit than the GLM, indicated by the instability of the results from different data folds of some of the variables. This indicates that the GBM has higher variance than the GLM, but also that it is less biased.

To identify any interactions detected by the GBM, we showed that the application of Friedman's H-statistic provides a concrete measure of the potential evidence of interactions. By applying Friedman's H-statistic for the trained GBM models we showed that we can compute the interaction strength for each of the explanatory variables in the data. The two-way H-statistics in Table 4.5 indicated that the actual interactions of the true frequency function also have the highest interaction strength. However, the two-way H-statistic also gave some evidence of interactions not actually included in the true frequency function. Once again, this is likely due to some overfitting done by the GBM. From some of the cross-validation results, seen in Appendix C, we saw that the selected models for some data folds have tree depths that allow for three-way interactions. Since we only have two-way interactions in the true frequency function these models will likely overfit.

By complementing the analysis of interaction effects with grouped partial dependence plots we saw that the true interactions showed significant differences in the claim frequency patterns (in Figure 4.6), while this was not as evident for the other interactions

(in Figure 4.7). Thus, the actual interactions included in the base case frequency function were also identified as the most likely candidates to be involved in any two-way interactions from the analysis of H-statistics and grouped partial dependence plots. It should be mentioned that the computation of the H-statistic can be quite demanding, depending on the number of explanatory variables, the size of the training data and the order of interaction effects computed. Computing the H-statistic for all two-way interactions in the base case simulation, i.e. with 50 000 observations and five explanatory variables, took about three hours. It is not uncommon for datasets the size of millions of observations in insurance and with many more variables, meaning long time for computations.

In short, the GBM model provided us with strong evidence for the actual interactions included in the true frequency function; we included these interactions in the GLMs (as GLM2 and GLM3); by doing that got improved model performance results over GLM1. Thus, in our base case scenario, we have shown that one can quite easily create an improved GLM based on the insights provided by the GBM. However, of course we cannot disregard the fact that we know the true frequency function and for this reason are biased. The interactions in the base case scenario are also very simple, making them easy to include in the GLM. In the case of more complex non-linearities or interactions it could be harder to improve the GLM, as indicated by one of the alternative scenarios.

From the alternative scenarios we saw that the performance of the GBM was significantly impacted by the number of simulated observations. The GBM performed worse than the GLM for smaller data sets, as seen in Tables 4.8 and 4.9. The reason for this is likely related to that changes in the data lead to very different series of splits in the trees of the GBM, thus leading to high variance. This is a common problem with tree-based methods. Therefore, it seems that the GBM is more prone to overfit if the data is noisy compared to the GLM which seems more robust. In the simulations, we started to see a significant performance improvement of applying the GBM compared to the GLM for 10 000 simulated observations and higher. However, one remark is that a finer search grid of the hyperparameter space would likely have been beneficial for the performance of the GBM in the case of smaller number of simulated observations.

The analysis of introducing a more complex interaction in the true frequency function in Section 4.4.2, showed that the GBM still incorporated the interaction effect indicated by strong evidence of the interaction strength given by the H-statistics and patterns according to grouped partial dependence plots. The result also indicated that the GLM did not work well for modeling the modified interaction. GBMs are known to be able to model complex non-linearities and interactions, which is confirmed by this result.

The scenario of applying the negative-binomial distribution for simulating the claims, gave similar results as the base case simulation but with an overall increase of the deviance and estimation loss for all models as seen in Tables 4.13 and 4.14. As mentioned, an increase in the performance measures was expected since the negative binomial distribution introduces more variance in the claims data generating process. However, our results shows that the performance impact on the GBM was no worse than for the GLM.

5.2 Drawbacks and limitations

Below are the known limitations in this thesis.

- **Simple simulation models.** Although the simulation models used in this thesis were inspired by real data, they were quite simplified. Mainly this was in order to allow easy comparisons of the fitted models, but also because of time constraints for writing the thesis and limitations in computing power. It would have been interesting to construct a more complex simulation model being closer to an actual true data generating process, including many more explanatory variables of different types, higher than second order interactions and more non-linear effects. Properly trained, we believe the GBM would show an even more obvious performance boost over the GLM in this case.

- **Small training datasets.** The largest simulated data sets were 50 000 observations in order to limit execution times of training the models, which in the context of pricing in insurance is rather small. We believe that larger datasets would have been especially helpful to deal with the variance problem of the GBMs.

- **No smoothing of effects for the GLMs.** We also acknowledge that our binning procedure of the policyholder age variable for the GLM likely could have been improved further, to make the comparison of the GBM with GLM more fair. Combining GAM with GLM to get smoother effects would likely have been beneficial in terms of GLM model performance, but we do not believe this would have been material enough to change any of the main results of this thesis. The main reason for not implementing a more sophisticated binning procedure was once again related to time constraints.

- **Limited search grid for hyperparameters and naive grid search strategy for the GBMs.** On the other hand, the GBMs were not really optimized with regards to their implementation either. In order to not get excessively long execution times for training the GBM models, we had to limit our search grid when tuning the hyperparameters. It is very likely we would have found better performing GBM models by using a finer and larger search grid. Besides the number of trees fitted and tree depth, we also could have chosen to tune some of the other hyperparameters in the GBM to get better performing models. Further, one could apply a more sophisticated method for tuning the hyperparameters than the cartesian grid search strategy we applied. There are methods available such as random search and Bayesian optimization which should be helpful in finding the best set of hyperparameters quicker than cartesian grid search.

5.3 Future work

Below are some suggestions for future work:

- **Address the limitations in Section 5.2.** This could imply using more complex simulation models, simulating larger datasets, applying improved fitting of the GLMs, using a larger and finer grid search and more sophisticated grid search strategies.

- **Investigate other distribution assumptions and loss functions for claim frequency.** One could furhter investigate the case of negative binomial distributed claims data or look into other claims distributions. Related to this, it would also be interesting to investigate the use of other loss functions for the GBM. For example, one could apply a negative binomial deviance loss function if one suspects that the data follows a negative binomial distribution.

- **Investigate claim severity and risk premium prediction with GBMs.** We have focused on the claim frequency part of non-life insurance pricing. It would also be interesting to make further investigations into claim severity and/or risk premium prediction.

- **Apply other tools for interpretation of ML-algorithms.** In terms of

interpreting the GBMs we have focused on global interpretation, i.e. to understand how the GBM models work from a global perspective. However, one could also look into methods that allow more local interpretation. Local interpretation methods explain why an individual prediction was made for a certain observation. In this area, methods such as LIME (Local Interpretable Model-agnostic Explanations) and Shapley values could be looked into. These type of methods could perhaps be helpful in explaining why a certain premium was given to a certain policyholder, if applying ML-algorithms such as GBMs for pricing.

CHAPTER 6
CONCLUSION

In our study we have shown that by using GBM models with a Poisson deviance loss function and by tuning of the model hyperparameters, we are able to create highly competitive models of claim frequency in a non-life insurance pricing setting. Given that we do not know about the interactions of our true frequency function, we have also shown that the GBM outperforms the GLM in terms of both prediction accuracy and ranking of the claim frequency risk.

The results of this simulation study confirms other studies in that GBM models have the advantage compared to GLMs that they are able to automatically detect and model non-linear effects and interaction effects between explanatory variables. For GLMs, such non-linear effects or interaction effects need to be manually investigated and included in the modeling which often is a time consuming effort. On the other hand, we saw that GLMs are relatively stable to fit whereas the GBMs are more prone to overfitting the data. Further, GLMs are quicker to train compared to GBMs. Finding a suitable GBM require cross-validation of a range of models to find a model with the best set of hyperparameters. Another obvious advantage of GLMs are their immediate transparency, since the parameter estimates of a GLM directly shows the effect of a variable on the response.

By using variable importance, partial dependence plots and Friedman's H-statistic we have also shown that we can gain an understanding of how the GBM models work and how we can extract insights from them. The application of these tools provide a good way of interpreting ML-algorithms such as GBMs, providing transparency and a way of explaining how the model works. Further, the GBM guided us in the right direction of creating GLMs with improved model performance.

The simulation models of claim frequency in this thesis were quite simple, having only five explanatory variables and two interactions. Real data is usually much more complex. In non-life insurance pricing, most larger insurance companies have access to

millions of rows of data with sometimes hundreds of potential explanatory variables. Here the potential benefits of applying GBMs is even more obvious. Even if it would not be feasible to directly apply GBMs for pricing, e.g. because of potential difficulty of model implementation for the insurer, GBMs can be a great complement to GLMs in the actuarial toolbox. As we have shown in this thesis, there are many methods available to extract potential insights from them.

APPENDIX A
EXAMPLES OF R CODE

```r
# Overview of utilized packages
library(tidyverse)
library(lubridate)
library(caret)
library(h2o)
library(iml)
library(statmod)
library(MASS)
library(modelr)
```

Code A.1: Important packages utilized

```r
# For every data fold k the following code is executed,
# where the data frame "data" contains all simulated data

# the target variable, the number of claims
y_claims <- "claims"

# the explanatory variables
# agephc is here the binned variant of ageph
varsglm <- c("agephc",
             "nbrcoi",
             "size",
             "rental",
             "student")

# the log of exposure, for use as offset
exp_freq <- "exposure_log"

# subset training data for data fold k, i.e. D / D_k
D_subset_k <- data %>%
  filter(!fold == i) %>%
```

```
  mutate(fold_rebased = ifelse(fold > i, fold - 1, fold))

# subset test data for data fold k, i.e. D_k
D_k <- data %>%
  filter(fold == i)

# create h2o data frames
D_subset_k_h2o <- as.h2o(D_subset_k)
D_k_h2o <- as.h2o(D_k)

# train glm for D / D_k
freq_glm_fit <- h2o.glm(y = y_claims, x = varsglm,
                        interactions = inter,
                        interaction_pairs = interpair,
                        training_frame = D_subset_k_h2o,
                        fold_column = "fold_rebased",
                        family = "poisson",
                        link = "log",
                        offset_column = exp_freq,
                        lambda = 0,
                        compute_p_values = TRUE,
                        remove_collinear_columns = TRUE,
                        solver = "IRLSM")
```

Code A.2: Code example for training GLM for a data fold k with h2o

```
# For every data fold k the following code is executed,
# where the data frame "data" contains all simulated data

# the target variable, the number of claims
y_claims <- "claims"

# the explanatory variables
varsgbm <- c("ageph",
             "nbrcoi",
             "size",
             "rental",
             "student")

# the log of exposure, for use as offset
exp_freq <- "exposure_log"

# define search grid for hyperparameters
```

```
ntrees <- seq(from = 100, to = 2000, by = 100)
max_depth <- seq(from = 1, to = 5, by = 1)

gbm_hyp_params <- list(
  ntrees = ntrees,
  max_depth = max_depth,
  learn_rate = 0.01,
  sample_rate = 0.75)


# subset training data for data fold k, i.e. D / D_k
D_subset_k <- data %>%
  filter(!fold == i) %>%
  mutate(fold_rebased = ifelse(fold > i, fold - 1, fold))

# subset test data for data fold k, i.e. D_k
D_k <- data %>%
  filter(fold == i)

# create h2o data frames
D_subset_k_h2o <- as.h2o(D_subset_k)
D_k_h2o <- as.h2o(D_k)

# set minimum 1 % of observations for a terminal node
min_rows_k <- nrow(D_subset_k)*0.01

grid_name <- paste0(name,"_grid_subset_fold_",i)

# train models via grid search in h2o
grid_gbm <- h2o.grid(x = x, y = y,
                     training_frame = D_subset_k_h2o,
                     fold_column = "fold_rebased",
                     algorithm = "gbm",
                     distribution = "Poisson",
                     offset_column = exposure,
                     grid_id = paste0(name,"_grid_subset_fold_",i),
                     hyper_params = gbm_hyp_params,
                     min_rows = min_rows_k,
                     seed = seed)


# Sort the grid models by residual deviance
```

```
60 sorted_grid <- h2o.getGrid(paste0(name,"_grid_subset_fold_",i),
61                            sort_by = "residual_deviance", decreasing =
                                 FALSE)
62 grids[[i]] <- sorted_grid
63 names(grids)[[i]] <- paste0(name,"_grid_subset_fold_",i)
64
65 best_model <- h2o.getModel(sorted_grid@model_ids[[1]])
```

Code A.3: Code example for training GBM for a data fold k with h2o

APPENDIX B
REGRESSION OUTPUT

name	coefficient	SE(σ)	z-score	p-value
Intercept	-1.05	0.03	-34.25	0.00
agephc.[20-24)	0.09	0.03	2.82	0.00
agephc.[24-28)	-0.03	0.03	-0.94	0.35
agephc.[28-31)	-0.26	0.03	-7.53	0.00
agephc.[31-34)	-0.51	0.04	-13.65	0.00
agephc.[34-37)	-0.79	0.04	-18.69	0.00
agephc.[37-41)	-1.13	0.05	-24.33	0.00
agephc.[41-45)	-1.53	0.06	-24.09	0.00
agephc.[45-50)	-2.11	0.10	-22.17	0.00
agephc.[50-55)	-2.59	0.17	-14.85	0.00
agephc.[55-58)	-2.51	0.33	-7.50	0.00
agephc.[58-100)	-3.84	0.71	-5.43	0.00
rental.Y	-0.61	0.02	-34.79	0.00
student.Y	-0.14	0.02	-7.41	0.00
nbrcoi	0.57	0.01	55.65	0.00
size	0.01	0.00	15.62	0.00

Table B.1: Regression output from the GLM fitted in data fold 1 of the base case simulation

name	coefficient	SE(σ)	z-score	p-value
Intercept	-1.03	0.03	-33.80	0.00
agephc.[20-24)	0.06	0.03	1.96	0.05
agephc.[24-28)	-0.06	0.03	-2.11	0.03
agephc.[28-31)	-0.28	0.03	-8.32	0.00
agephc.[31-34)	-0.52	0.04	-14.05	0.00
agephc.[34-37)	-0.82	0.04	-19.44	0.00
agephc.[37-41)	-1.13	0.05	-24.44	0.00
agephc.[41-45)	-1.59	0.07	-24.32	0.00
agephc.[45-50)	-2.10	0.09	-22.15	0.00
agephc.[50-55)	-2.56	0.17	-15.50	0.00
agephc.[55-58)	-2.44	0.32	-7.67	0.00
agephc.[58-100)	-3.80	0.71	-5.37	0.00
rental.Y	-0.61	0.02	-35.26	0.00
student.Y	-0.12	0.02	-6.42	0.00
nbrcoi	0.57	0.01	54.66	0.00
size	0.01	0.00	15.95	0.00

Table B.2: Regression output from the GLM fitted in data fold 2 of the base case simulation

name	coefficient	SE(σ)	z-score	p-value
Intercept	-1.02	0.03	-33.42	0.00
agephc.[20-24)	0.05	0.03	1.62	0.11
agephc.[24-28)	-0.08	0.03	-2.52	0.01
agephc.[28-31)	-0.29	0.03	-8.70	0.00
agephc.[31-34)	-0.55	0.04	-14.97	0.00
agephc.[34-37)	-0.85	0.04	-20.20	0.00
agephc.[37-41)	-1.19	0.05	-25.50	0.00
agephc.[41-45)	-1.65	0.07	-25.30	0.00
agephc.[45-50)	-2.18	0.10	-22.59	0.00
agephc.[50-55)	-2.67	0.17	-15.54	0.00
agephc.[55-58)	-2.63	0.35	-7.41	0.00
agephc.[58-100)	-3.86	0.71	-5.46	0.00
rental.Y	-0.60	0.02	-34.58	0.00
student.Y	-0.15	0.02	-7.97	0.00
nbrcoi	0.58	0.01	55.86	0.00
size	0.01	0.00	15.68	0.00

Table B.3: Regression output from the GLM fitted in data fold 3 of the base case simulation

name	coefficient	SE(σ)	z-score	p-value
Intercept	-1.03	0.03	-33.72	0.00
agephc.[20-24)	0.06	0.03	1.98	0.05
agephc.[24-28)	-0.06	0.03	-2.06	0.04
agephc.[28-31)	-0.27	0.03	-8.06	0.00
agephc.[31-34)	-0.53	0.04	-14.25	0.00
agephc.[34-37)	-0.83	0.04	-19.61	0.00
agephc.[37-41)	-1.19	0.05	-25.47	0.00
agephc.[41-45)	-1.63	0.07	-24.87	0.00
agephc.[45-50)	-2.09	0.09	-22.62	0.00
agephc.[50-55)	-2.74	0.19	-14.80	0.00
agephc.[55-58)	-2.30	0.29	-7.91	0.00
agephc.[58-100)	-4.53	1.00	-4.53	0.00
rental.Y	-0.61	0.02	-35.22	0.00
student.Y	-0.14	0.02	-7.22	0.00
nbrcoi	0.58	0.01	55.88	0.00
size	0.01	0.00	15.59	0.00

Table B.4: Regression output from the GLM fitted in data fold 5 of the base case simulation

name	coefficient	SE(σ)	z-score	p-value
Intercept	-1.02	0.03	-33.56	0.00
agephc.[20-24)	0.04	0.03	1.31	0.19
agephc.[24-28)	-0.07	0.03	-2.36	0.02
agephc.[28-31)	-0.29	0.03	-8.68	0.00
agephc.[31-34)	-0.53	0.04	-14.39	0.00
agephc.[34-37)	-0.83	0.04	-19.62	0.00
agephc.[37-41)	-1.16	0.05	-24.98	0.00
agephc.[41-45)	-1.64	0.07	-24.99	0.00
agephc.[45-50)	-2.08	0.09	-22.70	0.00
agephc.[50-55)	-2.70	0.18	-15.07	0.00
agephc.[55-58)	-2.48	0.33	-7.42	0.00
agephc.[58-100)	-4.60	1.00	-4.60	0.00
rental.Y	-0.62	0.02	-35.40	0.00
student.Y	-0.13	0.02	-6.49	0.00
nbrcoi	0.58	0.01	55.84	0.00
size	0.01	0.00	15.64	0.00

Table B.5: Regression output from the GLM fitted in data fold 6 of the base case simulation

APPENDIX C
GBM CROSS-VALIDATION OUTPUT

number of trees (M)	tree depth (d)	model id	residual deviance
1500	3	fold_1_model_73	1.190750
1600	3	fold_1_model_78	1.190752
1700	3	fold_1_model_83	1.190752
1800	3	fold_1_model_88	1.190752
1900	3	fold_1_model_93	1.190752
2000	3	fold_1_model_98	1.190752
1400	3	fold_1_model_68	1.190769
1300	3	fold_1_model_63	1.190803
1200	3	fold_1_model_58	1.190864
1700	2	fold_1_model_82	1.190926

Table C.1: Grid search cross-validation results of the 10 best performing models for data fold 1 of the base case simulation. For all models the learn rate (τ) = 0.01, sample rate (δ) = 0.75 and min observations (κ) = 0.01.

number of trees (M)	tree depth (d)	model id	residual deviance
1700	2	fold_2_model_82	1.191696
1800	2	fold_2_model_87	1.191696
1900	2	fold_2_model_92	1.191696
2000	2	fold_2_model_97	1.191696
1600	2	fold_2_model_77	1.191713
1500	2	fold_2_model_72	1.191790
1900	3	fold_2_model_93	1.191830
2000	3	fold_2_model_98	1.191830
1800	3	fold_2_model_88	1.191834
1400	2	fold_2_model_67	1.191846

Table C.2: Grid search cross-validation results of the 10 best performing models for data fold 2 of the base case simulation. For all models the learn rate (τ) = 0.01, sample rate (δ) = 0.75 and min observations (κ) = 0.01.

number of trees (M)	tree depth (d)	model id	residual deviance
1900	3	fold_3_model_93	1.189089
2000	3	fold_3_model_98	1.189089
1800	3	fold_3_model_88	1.189099
1600	3	fold_3_model_78	1.189106
1700	3	fold_3_model_83	1.189107
1700	2	fold_3_model_82	1.189126
1800	2	fold_3_model_87	1.189126
1900	2	fold_3_model_92	1.189126
2000	2	fold_3_model_97	1.189126
1500	3	fold_3_model_73	1.189127

Table C.3: Grid search cross-validation results of the 10 best performing models for data fold 3 of the base case simulation. For all models the learn rate (τ) = 0.01, sample rate (δ) = 0.75 and min observations (κ) = 0.01.

number of trees (M)	tree depth (d)	model id	residual deviance
1700	2	fold_5_model_82	1.189655
1800	2	fold_5_model_87	1.189655
1900	2	fold_5_model_92	1.189655
2000	2	fold_5_model_97	1.189655
1600	2	fold_5_model_77	1.189672
1500	2	fold_5_model_72	1.189690
1400	2	fold_5_model_67	1.189756
1300	2	fold_5_model_62	1.189818
2000	3	fold_5_model_98	1.189959
1900	3	fold_5_model_93	1.189970

Table C.4: Grid search cross-validation results of the 10 best performing models for data fold 5 of the base case simulation. For all models the learn rate (τ) = 0.01, sample rate (δ) = 0.75 and min observations (κ) = 0.01.

number of trees (M)	tree depth (d)	model id	residual deviance
2000	3	fold_6_model_98	1.189467
1900	3	fold_6_model_93	1.189471
1500	3	fold_6_model_73	1.189472
1800	3	fold_6_model_88	1.189477
1700	3	fold_6_model_83	1.189484
1600	3	fold_6_model_78	1.189489
1400	3	fold_6_model_68	1.189523
1600	2	fold_6_model_77	1.189526
1700	2	fold_6_model_82	1.189528
1800	2	fold_6_model_87	1.189528

Table C.5: Grid search cross-validation results of the 10 best performing models for data fold 6 of the base case simulation. For all models the learn rate (τ) = 0.01, sample rate (δ) = 0.75 and min observations (κ) = 0.01.

REFERENCES

10 Anderson, D., Feldblum, S., Modlin, C., Schirmacher, D., Schirmacher, E., & Thandi, N. (2004). A practitioner's guide to generalized linear models. *Casualty Actuarial Society Discussion Paper Program*, 1–116.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Dionne, G., Gouriéroux, C., & Vanasse, C. (1999). Evidence of adverse selection in automobile insurance markets. In *Automobile insurance: Road safety, new drivers, risks, insurance fraud and regulation* (pp. 13–46). Springer.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378.

Friedman, J. H., Popescu, B. E., & others. (2008). Predictive learning via rule ensembles. *Annals of Applied Statistics*, *2*(3), 916–954.

Fujita, S., Tanaka, T., & Iwasawa, H. (2020). AGLM: A hybrid modeling method of GLM and data science techniques.

Goldburd, M., Khare, A., & Tevet, D. (2016). Generalized linear models for insurance rating. *Casualty Actuarial Society, CAS Monographs Series*, *5*.

Guelman, L. (2012). Gradient boosting trees for auto insurance loss cost modeling and prediction. *Expert Systems with Applications*, *39*(3), 3659–3667.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. Springer Science & Business Media.

Henckaerts, R., Côté, M.-P., Antonio, K., & Verbelen, R. (2020). Boosting insights in insurance tariff plans with tree-based machine learning methods. *North American Actuarial Journal*, 1–31.

Kaminski, M. E. (2019). The right to explanation, explained. *Berkeley Tech. LJ*, *34*, 189.

Lee, S. C., & Lin, S. (2018). Delta boosting machine with application to general insurance. *North American Actuarial Journal*, *22*(3), 405–425.

Nelder, J. A., & Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, *135*(3), 370–384.

Ohlsson, E., & Johansson, B. (2010). Non-life insurance pricing. In *Non-life insurance pricing with generalized linear models* (pp. 1–14). Springer.

Wuthrich, M. V., & Buser, C. (2019). Data analytics for non-life insurance pricing. *Swiss Finance Institute Research Paper*, (16-68).

Yang, Y., Qian, W., & Zou, H. (2018). Insurance premium prediction via gradient tree-boosted tweedie compound poisson models. *Journal of Business & Economic Statistics*, *36*(3), 456–470.

Zhou, H., Qian, W., & Yang, Y. (2020). Tweedie gradient boosting for extremely unbalanced zero-inflated data. *Communications in Statistics-Simulation and Computation*, 1–23.

Zöchbauer, P., Wüthrich, M. V., & Buser, C. (2016). Data science in non-life insurance pricing. *Master's Thesis ETH Zürich, Zürich*.