EXAMENSARBETE INOM TEKNIK,
GRUNDNIVÅ, 15 HP

STOCKHOLM, *SVERIGE 2020*

Tree-based Machine Learning Models

with Applications in Insurance Frequency Modelling

SAMUEL TOBER

KTH
SKOLAN FÖR TEKNIKVETENSKAP

Abstract

As the insurance industry is highly data driven it is no surprise that machine learning (ML) has made its way into the industry. While GLMs are still the comfort zone of most actuaries, we have in recent years seen a surge in machine learning algorithms. This study puts focus on developing and evaluating three tree-based machine learning models, starting from simple decision trees and working up to the more advanced ensemble methods random forests and gradient boosting machines. We predict the claims frequency for an all-risk insurance tariff through a case study based on a data set provided by a Swedish insurance company. The gradient boosting machines and random forests are found to outperform the single decision trees, and moreover, we use visualisation tools to uncover and gain insights from the models.

Key Words

Decision trees, machine learning, frequency modelling, insurance data set, cross-validation

Sammanfattning

Med tanke på att försäkringsbranschen i stor utsträckning förlitar sig på stora mängder data, är det inte förvånande att maskininlärningstekniker har börjat göra avtryck på industrin. Fastän det flesta aktuarier fortfarande jobbar med GLM modeller, har vi på senare år sett ett uppsving av maskininlärning inom försäkringsmodellering. Denna studie ämnar att utveckla och utvärdera tre trädbaserade maskininlärningsmodeller, från ett enkelt beslutsträd till de mer avancerade ensemble metoderna random forest och gradient boosting machines. Vi modellerar skadefrekvensen för en allrisk försäkring genom en fallstudie på försäkringsdata från ett svenskt försäkringsbolag. Gradient boosting machines och random forest visar sig överträffa de simpla beslutsträden, och vidare använder vi visualiseringsmetoder för att tolka och få insikter från modellerna.

Nyckelord

Beslutsträd, maskininlärning, frekvensmodellering, försäkringsdata, korsvalidering

Acknowledgements

I would like to express my very great appreciation to Henrik Bosaeus for valuable and constructive suggestions during the course of this research work. His willingness to give his time so generously has been very much appreciated. My grateful thanks are also extended to Daniel Berglund for his patient guidance and advice. Special thanks should also be given to Roel Henckaerts for his valuable recommendations on the implementation details of tree-based models in R.

Contents

1 Introduction

Due to technological breakthroughs and the realisation of big data in many industries, the popularity of using machine learning (ML) in business application has seen surge, and the insurance industry is no exception. Although most research in the field of insurance modelling is still focused on so called generalised linear models, there are notable papers in which ML is used in rate making. Wütrich and Buser show how ML models can be used to provide estimates for frequency in an insurance context [5]. Henckaerts et al. predict both frequency and severity from an insurance portfolio, using tree based ML methods, and shows how these methods can surpass the traditional GLM [8]. This paper will largely follow in their footsteps, in hope of validating their results through a case study of a non-life insurance tariff, but also to bring insights to a Swedish insurer who's data serves as the foundation for the models in this study. Moreover, this study differs in that it focuses on a home insurance tariff rather than auto insurance. This paper is outlined as follows: First, an overview of the business model and common practices of insurance companies will be given. Second, the fundamentals of tree-based ML models will be introduced. Third, we will apply this knowledge in the context of a case study, with the purpose of evaluating and comparing the ML models.

2 Insurance Fundamentals

This section aims to, from a broad perspective, explain the industry of insurance and the current state-of-the-art modelling techniques employed in this domain.

2.1 Underlying Principles of Insurance

The insurance industry is an industry concerned with hedging against the risk of uncertain financial loss, and the business of insurance companies is therefore largely a risk management endeavour. The insured trades future risk with an insurer for a fixed premium through a contract, known as the insurance policy, and if the policy holder is subject to a

loss they can submit a claim to the insurer, if permitted by the policy. The premium is set by the insurer in advance of any claims, and hence it is vital for the company to predict the risks of their customers in order to set a profitable premium. With this in mind, it is not surprising that predictive modelling is extensively used in insurance companies; both in assessing customers and setting premiums.

Insurance companies make money through both underwriting and by investing the premiums collected from customers, underwriting being the primary work carried out at the insurance company, involving measuring the risk of insuring a customer and what premium should be charged in order to make a profit [1]. Any risk that can be quantified can potentially be insured and specific kinds of risks that can give rise to claims are known as perils in the insurance policy. For example, if we consider a home insurance policy in the US, usually, damage to the home and the owner's belongings are both covered by the policy. In contrast, a Swedish home insurance always includes: movables (covers theft, fire damage etc...), travel protection, liability protection, legal expenses and assault protection [2]. Moreover, if the insurance undertaker wishes they can usually purchase additional insurance, for example all-risk which covers all damage except specific perils that are excluded. Examples of excluded perils are: damage due to war, a flood or an earthquake.

2.2 Insurance Pricing

First, it is important to understand why setting a premium that rightfully corresponds to the risk of the customer is so crucial for the insurer. We will highlight this with a simple example. Assume two insurers, A and B, exist and A has a low premium relative to the risk of loss, while B has an adequate premium in relation to the risk. In this scenario, a high risk customers would opt for A since their premium is relatively low compared to B, thus A would attract high risk customers and in effect see their margins being eaten up. On the contrary, if A's premiums are too high they would not attract any profitable

customers and still lose money. In the light of this simple example, we see why a competitive pricing strategy is paramount. Perhaps, the most common pricing strategy is the frequency-severity strategy.

Opting for the frequency-severity strategy, we assume there are two factors influencing the price [3]:

1. Frequency (F) - number of claims per exposure time
2. Severity (S) - average loss per claim

where exposure is the time for which a risk is insured. Now, assume that an insurer has a total loss of L spread out over N claims and an exposure e. Then the effective, or technical premium would be [4]:

$$\tau = \mathbb{E}\left(\frac{L}{e}\right) = \mathbb{E}\left(\frac{L}{N}|N > 0\right) \times \mathbb{E}\left(\frac{N}{e}\right) = \mathbb{E}\left(S\right) \times \mathbb{E}\left(F\right) \tag{1}$$

Where we have assumed independence between the frequency and severity. By this reduction, insurance pricing becomes a problem of predicting S and F. Be that as it may, in this paper we will restrict our study to the frequency.

2.3 Predictive Modelling in Insurance

In general, predicting the frequency can be scaled down to finding a function, or model, $f_F(\cdot)$, given features $\boldsymbol{x}$, such that:

$$F = f_F(\boldsymbol{x}) \tag{2}$$

where $\boldsymbol{x}$ is the training data of the model. In almost all cases this function $f_F(\boldsymbol{x})$ cannot be found explicitly. Instead, we try to approximate f_F as well as we can.

The modelling cycle

Insurance companies usually rely on statistical methods to find approximations for f_F. There are of course many ways to approach modelling, however, we can summarise the modelling cycle from a high-level perspective by the following six steps [5]:

The modelling cycle

1. Data collection, data cleaning, data pre-processing, data visualisation
2. Selection of model class and predictive variables
3. Choice of loss function
4. Solving an optimisation problem
    - Choosing optimisation algorithm
    - Choosing step length
    - Choosing stopping criteria
    - Choosing initial seed of the algorithm
5. Model validation
6. Possibly we have to move back to item 1. if we are not satisfied by the "solution"

If we are to look at these steps from an insurance companies perspective we can note a few things. First, insurance companies generally have an abundance of data, however, the data usually needs to be processed to provide useful insight. Second, the go-to model for most insurance companies is, and has been for the last 30 years, the generalised linear model (GLM), which generalises linear regression by allowing for non-linear relation between the predictive variables and the response via a so-called link function. GLMs can be formulated as:

$$\mathbb{E}(\boldsymbol{Y}) = g^{-1}(\boldsymbol{X} \cdot \boldsymbol{\beta}) \tag{3}$$

where $\boldsymbol{Y}$ is the response, $g(\cdot)$ the link function, $\boldsymbol{X}$ the predictive variables and $\boldsymbol{\beta}$ unknown parameters. This paper does not aim to give an exhaustive exposition of GLMs, if the reader is interested in reading more on the subject, please refer to [6] [7]. Be it implicitly or explicitly, essentially all models have underlying assumptions, and models used in rate making are no exception. Next, we will discuss the assumptions commonly made in frequency models and moreover, derive a useful metric to assess the fit of a model given these assumptions.

Model Assumptions for Claim Frequency Modelling

The first assumption we make is on the distribution of the number of claims. The number of claims, N, filed by a customer is usually assumed to follow a Poisson distribution [8]. A discrete random variable, N, is said to follow a Poisson distribution with parameters $\lambda \in \mathbb{R}$ and $v \in \mathbb{R}^+$, if for $k \in \mathbb{N}_0$, the probability mass function of X is given by:

$$f(k, \lambda v) = \mathbb{P}(N = k) = e^{-\lambda v} \frac{(\lambda v)^k}{k!} \tag{4}$$

where $\mathbb{E}(X) = \mathrm{Var}(X) = \lambda$. The volume v often measures the exposure in years. The second assumption we make is that the portfolio is homogeneous, meaning that we see the claims $\{N_1, N_2, ..., N_n\}$ as a family of independent Poisson distributed variables, with the same parameter λ for all N_i. In this way, the modelling problem becomes a problem of estimating the parameter λ. With this assumption we can write down the joint likelihood (equation 5) and log-likelihood (equation 6) for the claims vector $\boldsymbol{N} = \{N_1, N_2, ..., N_n\}$:

$$\ell_{\boldsymbol{N}}(\lambda) = \mathbb{P}(N_1 = k_1, ..., N_n = k_n) = \prod_{i=1}^{n} \mathbb{P}(N_i = k_i) = \prod_{i=1}^{n} e^{-\lambda v_i} \frac{(\lambda v_i)^{N_i}}{N_i!} \tag{5}$$

$$\log \ell_{\boldsymbol{N}}(\lambda) = \sum_{i=1}^{n} -\lambda v_i + N_i \log(\lambda v_i) - \log(N_i!) \tag{6}$$

Now, if we instead let every N_i have a corresponding λ_i and use the ML-estimate, $\lambda_i^{\mathrm{ML}} =$

$\frac{N_i}{v_i}$, we get the so called saturated model which has a log-likelihood of (by inserting λ_i^{ML} into equation 7):

$$\log \ell_{\boldsymbol{N}}^{s}(\boldsymbol{N}) = \sum_{i=1}^{n} -N_i + N_i \log(N_i) - \log(N_i!) \tag{7}$$

Taking the difference of the log-likelihood for the saturated model and the homogeneous model, and multiplying by a factor of 2, we arrive at what is called the Poisson deviance:

$$D(\boldsymbol{N}, \lambda) = 2[\log \ell_{\boldsymbol{N}}^{s}(\boldsymbol{N}) - \log \ell_{\boldsymbol{N}}(\lambda)] = 2 \sum_{i=1}^{n} [N_i \log \frac{N_i}{\lambda v_i} - (N_i - \lambda v_i)] \tag{8}$$

In words, the Poisson deviance is the difference between the prediction of the saturated model, which in a way is the maximally overfitted model, and the model of interest. The Poisson deviance is normally used to assess the goodness of fit in a Poisson regression context, where a high deviance means a poor fit and vice versa for a low deviance. Later, we will rely on this metric in evaluating all the models built in the case study, but first it is necessary to introduce the fundamental ML theory used in all models.

3 Machine Learning Fundamentals

In this section we present the basic concepts of ML used in building, tuning and evaluating ML models. First, we introduce three different models, all of which are based on the decision tree. Second, we discuss how these models' parameters can be optimally tuned. Third, we cover methods to uncover, and evaluate the models.

3.1 Decision Trees

One common type of ML model is the *decision tree*, introduced by Breiman et al. [9] in 1984, which is a very intuitive and natural model for us humans as it in a way mimics the way we make decisions. In order to give a formal definition of a decision tree, we first need to define predictor space.

Definition 3.1. The *predictor space*, $R(\boldsymbol{x})$, for a vector of continuous random variables $\boldsymbol{x} \in \mathbb{R}^p$, is the set of possible values for the p variables.

Definition 3.2. Given a predictor space $R(\boldsymbol{x})$, a *decision tree*, $f_T(\boldsymbol{x})$, is a predictive model that partitions R into J distinct, non-overlapping regions R_j, with a fitted response $\hat{y}_{R_j}$, such that:

$$f_T(\boldsymbol{x}) = \sum_{j=1}^{J} \hat{y}_{R_j} \mathbb{1}(\boldsymbol{x} \in R_j) \tag{9}$$

where,

$$R = \cup_{j=1}^{J} R_j \text{ s.t. } R_i \cap R_j = \emptyset, \quad \forall i \neq j$$

and,

$$\mathbb{1}(P) = \begin{cases} 1 & \text{if P is true} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Obviously the above definition is very general, and there are indeed many ways of growing a tree for both classification and regression problems, however the most commonly used algorithm is the Classification and Regression Tree algorithm (CART). One of the issues in growing the tree is that there is a plethora of ways to partition the predictor space, and therefore CART uses a greedy approach to the partitioning, known as recursive binary splitting. It works by asking a sequence of hierarchical questions starting from the so called root node, consisting of the entire training set. A node is a subset of features, and can be either terminal or non-terminal. A non-terminal node can split into two daughter nodes, and this binary split is determined by a condition on one variable, and once a value in the original set reaches this node it either satisfies the condition or not. If it does it goes down to one of the daughter nodes, if not it goes down to the other. A node that does not split is called a terminal node [10]. See Figure 1 for an example of a decision tree.

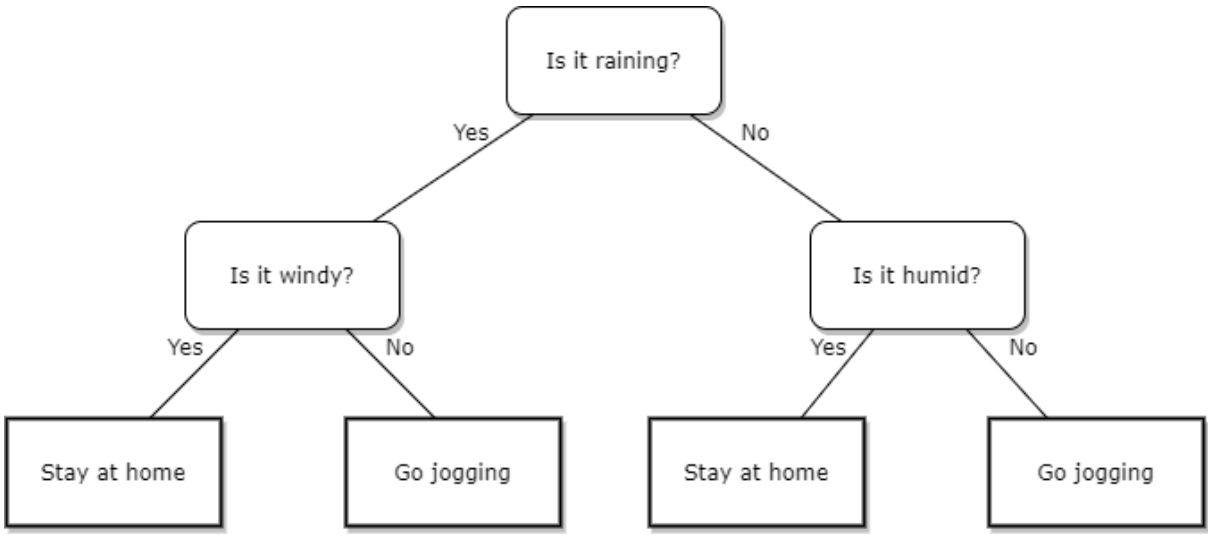

Figure 1: Example of a decision tree which decides if we should go jogging or not based on the weather

CART

CART consists of two steps - one growing step and one pruning step. Lets first consider a node t, a candidate for a split condition s and a variable X that the split is done on, CART then splits t into a right node, $t_R = \{X : X > s\}$, and a left node, $t_L = \{X : X \leq s\}$ so that $t = t_L \cup t_R$. This procedure is then recursively continued until a stopping criterion is reached, such as a maximum depth of the tree. Generally, s and X are chosen so that some loss function $\mathcal{L}(\cdot, \cdot)$ is minimised given the two daughter nodes [9]:

$$s, X = \underset{s,X}{\operatorname{argmin}} \left[\sum_{\boldsymbol{x_i} \in t_R(s,X)} \mathcal{L}(y_i, \hat{y}_{t_R}) + \sum_{\boldsymbol{x_i} \in t_L(s,X)} \mathcal{L}(y_i, \hat{y}_{t_L}) \right] \tag{11}$$

Loss function

In order to use CART, one first has to choose a loss function [10]. The most common choice of loss function in a regression context is the mean squared loss:

$$\mathcal{L}(y_i, \hat{y}_i) = \text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{12}$$

Though, one should always choose loss function depending on the distribution of the data

at hand. The MSE is appropriate if the data is normally distributed but, for example if the data is distributed according to a Poisson distribution, the MSE might not be the best option [8]. In fact, the Poisson deviance we derived earlier (equation 8) is a much better choice as it rests on the assumption of a Poisson distributed predictor variable. Therefore it will be used as the loss function in all decision trees built in this paper.

Bias-Variance Trade-Off

So far we have only discussed how to grow the tree, but as mentioned above there are two steps to CART. Before we go into explaining the next step we need to have an understanding of an important concept in modelling, namely, the *bias-variance trade off*. Using decision trees as an example, imagine we grow a tree so deep that every single input value has a corresponding terminal node. In this way, every data point in the training set would be correctly classified and we say that the model has very low *bias* on the training data. Now, if we use this model to predict the response of a new input, as one can imagine, the results would vary a lot depending on the input. In other words, the model has high *variance*. In this case, we say that the tree is *overfitted* on the training set. On the other end of the spectrum, we could grow a very short tree, which would be more biased on the training set, but hopefully generalise better to new data. This is known as the bias-variance trade off; as the complexity of the model increases, we reduce bias at the cost of an increased variance and vice versa. This phenomenon can be summarised by Figure 2. [11]

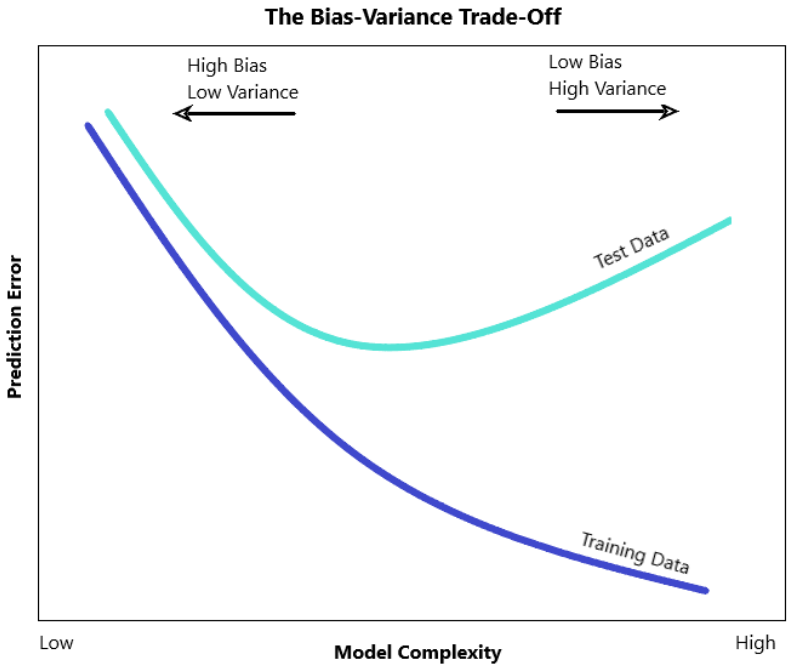

Figure 2: In all ML models there is a trade-off between bias and variance

In light of the previous discussion, we now realise that if the tree is grown deep we risk overfitting the model. Therefore, CART does not only make sure the model fits the data, it also penalises a tree that is too complex. In practice this is done by introducing a complexity parameter, cp, in the minimisation:

$$\sum_{j=1}^{J}\left[\sum_{\boldsymbol{x_i}\in R_j}\mathcal{L}(y_i, y_{\hat{R}_j})\right] + \text{cp}\cdot J\sum_{\boldsymbol{x_i}\in R}\mathcal{L}(y_i,\hat{y}_R) \tag{13}$$

where the first term ensures a good fit and the second reduces overfitting according to the constant cp, with cp $= 1$ resulting in a tree without splits and cp $= 0$ a maximally deep tree. One of the first caveats we face is how we choose an optimal value of cp. This is most commonly done through a technique known as cross validation. [12]

3.2 Cross-Validation

In k-fold cross validation we divide a dataset, $\mathcal{D}$, into k mutually exclusive sets $\mathcal{D} = \bigcup_{i=1}^{k}\mathcal{D}_i$, these sets are iterated over, and in each iteration one of the sets, $\{\mathcal{D}_i : i \in$

$\{1, 2, ..., k\}\}$, is left out for subsequent testing. Using the data left after leaving out a test set, $\mathcal{D} \setminus \mathcal{D}_i$, we can again iteratively leave out one of the data sets, $\{\mathcal{D}_l : l \in \{1, 2, ..., k\} \setminus \{i\}\}$, train a tree on the remaining data, $\mathcal{D} \setminus \{\mathcal{D}_i, \mathcal{D}_l\}$, validate on $\mathcal{D}_l$, use the parameters which minimise the average validation error, build a model with these parameters and finally test it on $\mathcal{D}_i$. In the end, we get results consisting of the test error for the k different folds. This procedure, with $k = 6$, is illustrated in Figure 3. In this way we completely utilise the data at hand, in contrast to for example dividing all data into one test set and one training set, this method lets us get the most out of our data. This technique can be used to tune parameters such as the complexity parameter in the decision tree.

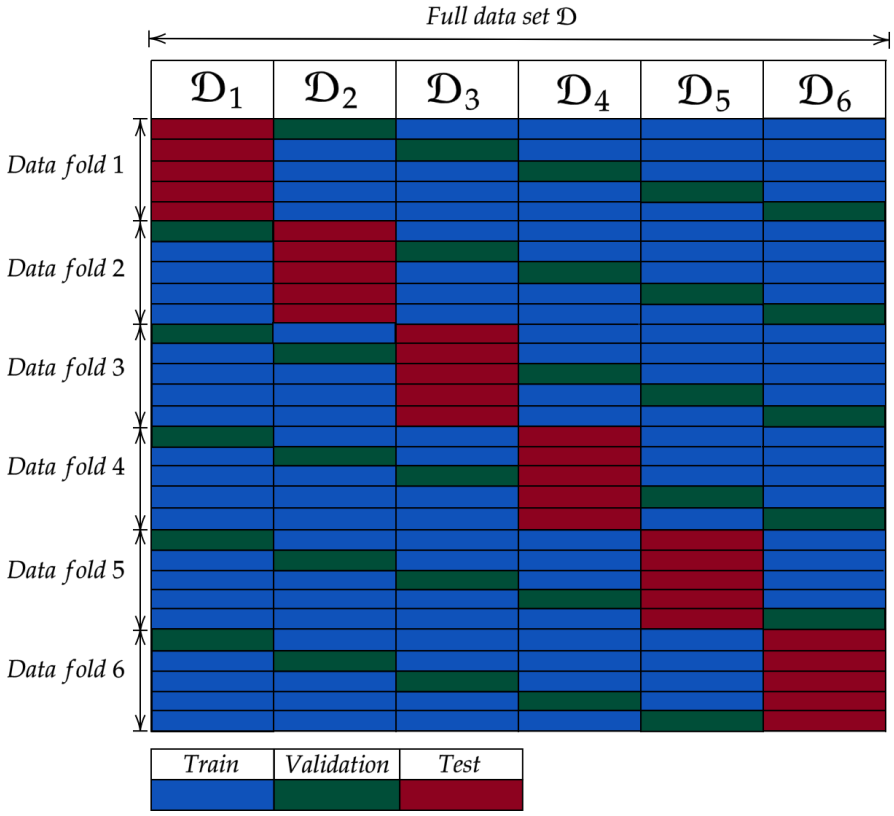

Figure 3: The data set is partitioned into 6 folds, one fold is left out for testing, and 5-fold cross-validation is iteratively performed on the remaining data to find the optimal model parameters for the current fold

3.3 Ensemble Methods

There are obvious advantages of decision trees, such as their interpretability and the fact that they can combine both continuous and discrete data. However, they also have their limitations. For one, single decision trees tend to have a rather high variance and can be very sensitive to the training data [13]. In order to counteract this shortcoming, so called ensemble methods can be used, in which multiple weak models are aggregated into a more powerful predictor. There are many ensemble techniques, however in the scope of this paper, three methods will be covered: bagging, random forests and gradient boosting machines.

Bagging

Bootstrap aggregating, or bagging, was introduced in 1996 by Leo Breiman [14]. In short it is a method of aggregating several decision trees into a single predictor and thereafter predicting a response based on averaging for regression problems, and majority vote for classification problems. More specifically, given a training set $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^{N}$, where y_i are the labels and $\boldsymbol{x_i}$ the features, and a modelling procedure $\hat{y} = f(\boldsymbol{x}, \mathcal{D})$, we train T models $\{f_i\}_{i=1}^{T}$ on T different bootstrap samples, $\mathcal{L}_i$, of $\mathcal{D}$. If the problem at hand is a regression problem we find the prediction by $\frac{1}{T}\sum_{i=1}^{T} f_i(\boldsymbol{x}, \mathcal{L}_i)$. While bagging gives improvement compared to single decision trees, it has a significant problem. Namely, the splits in each decision tree will be based on what variables minimise the loss function the most, and therefore the T trees will be somewhat correlated, in that they will most likely split on the same variables. This problem is addressed in the next technique, random forests.

Random forests

Random forests are identical to bagging, with the main difference being in the variable selection, which in random forests, is done by random sampling from all features to ensure that trees wont be too similar. That is, when growing the decision trees not necessarily all

features are used [15]. See Algorithm 1 for more details on implementation of the random forest model.

Algorithm 1 Random Forest

for *i=1 to T* **do**
  generate random sample $\mathcal{L}_i$
  while *stopping criteria not satisfied* **do**
    - randomly select m of all features
    - train a tree on these features
  end
end
$f_{\text{rf}}(\boldsymbol{x}) = \frac{1}{T}\sum_{i=1}^{T} f_{\text{tree}}(\boldsymbol{x}|\mathcal{L}_i)$

Gradient Boosting

J. Friedman introduced gradient boosting in his 1999 paper [16]. The general problem of predictive modelling is, as we now know, to find a function $f(\boldsymbol{x})$ to predict a response variable $\boldsymbol{y}$ from a set of explanatory variables $\boldsymbol{x}$, which minimises some loss function $\mathcal{L}(f(\boldsymbol{x}), y)$. Gradient boosting is considered a gradient descent algorithm, meaning it relies on iterative tuning of parameters in order to achieve the minimium of a specified loss function. In boosting, $f(\boldsymbol{x})$ is estimated by an expansion of the form:

$$\hat{f}(\boldsymbol{x}) = \sum_{m=0}^{M} \beta_m h(\boldsymbol{x}, \boldsymbol{a}_m) \tag{14}$$

where the base learners $h(\boldsymbol{x}, \boldsymbol{a}_m)$ are usually chosen to be simple functions with parameters $\boldsymbol{a}$. Both $\boldsymbol{a}$ and β are fitted to the training data in a step-wise manner. We start with an initial guess $\hat{f}_0(\boldsymbol{x})$ and then for each m we evaluate:

$$\begin{aligned} (\beta_m, \boldsymbol{a}_m) &= \underset{\beta, \boldsymbol{a}}{\operatorname{argmin}} \sum_{i=0}^{N} \mathcal{L}(y_i, \hat{f}_{m-1}(\boldsymbol{x}_i) + \beta h(\boldsymbol{x}_i, \boldsymbol{a})) \\ \hat{f}_m(\boldsymbol{x}) &= \hat{f}_{m-1}(\boldsymbol{x}) + \beta_m h(\boldsymbol{x}, \boldsymbol{a}_m)) \end{aligned} \tag{15}$$

Friedmans gradient boosting approximately solves Equation 15 through a two step pro-

cedure. First, $h(\boldsymbol{x}, \boldsymbol{a})$ is fit by least-squares:

$$\boldsymbol{a}_m = \underset{\boldsymbol{a},\rho}{\operatorname{argmin}} \sum_{i=0}^{N} [\tilde{y}_{im} - \rho h(\boldsymbol{x}_i, \boldsymbol{a})]^2 \tag{16}$$

where,

$$\begin{aligned} \tilde{y}_{im} &= -\left[\frac{\partial L(y_i, \hat{f}(\boldsymbol{x}_i))}{\partial \hat{f}(\boldsymbol{x}_i)}\right] \\ \hat{f}(\boldsymbol{x}) &= \hat{f}_{m-1}(\boldsymbol{x}) \end{aligned} \tag{17}$$

and then β_m is determined by:

$$\beta_m = \underset{\beta}{\operatorname{argmin}} \sum_{i=0}^{N} \mathcal{L}(y_i, \hat{f}_{m-1}(\boldsymbol{x}_i) + \beta h(\boldsymbol{x}_i, \boldsymbol{a}_m)) \tag{18}$$

The choice of the base learners $h(\cdot, \cdot)$ is arbitrary, and naturally we choose to use decision trees in this study. In this setting, the parameters $\boldsymbol{a}_m$ are the splitting variables and splitting points defining the tree. The base learner then becomes:

$$h(\boldsymbol{x}_i, \{R_{lm}\}_1^L) = \sum_{l=1}^{L} \overline{y}_{lm} \mathbb{1}(\boldsymbol{x} \in R_{lm}) \tag{19}$$

with $\overline{y}_{lm}$ being the mean of $\tilde{y}_{im}$ in the region R_{lm}. Since the value of $h(\cdot, \cdot)$ is constant in each region of the tree Equation 18 simplifies to:

$$\gamma_{lm} = \underset{\gamma}{\operatorname{argmin}} \sum_{i=0}^{N} \mathcal{L}(y_i, \hat{f}_{m-1}(\boldsymbol{x}_i) + \gamma) \tag{20}$$

Finally, the current approximation $\hat{f}(\boldsymbol{x})_{m-1}$ is updated for each region R_{lm}, using γ_{lm}:

$$\hat{f}_m(\boldsymbol{x}) = \hat{f}_{m-1}(\boldsymbol{x}) + \lambda \gamma_{lm} \mathbb{1}(\boldsymbol{x} \in R_{lm}) \tag{21}$$

where a shrinkage parameter, $0 < \lambda \leq 1$, which determines the learning rate of the algorithm, is introduced.

We have now covered all the models that will be used in the subsequent case study, and now move on by presenting methods to evaluate these models.

3.4 Evaluation Methods

According to the European Union General Data Protection Regulation (GDPR), in the context of automated decision-making [17]:

> "[*the data subject should have*] *the right ... to obtain an explanation of the decision reached.*"

Therefore, it is important to be able to understand, interpret and evaluate the models we build. In doing this we will utilise two different measures: Variable importance and partial dependency.

Variable importance

Variable importance measures how important the independent variables are in predicting the dependent variable. The measure was introduced by Breiman in 2001 [15] and he defined importance of a variable in terms of decrease in the loss function when the variable is chosen as the feature to split a node on. It can be written as a sum over all the splits where the variable of interest is involved:

$$\mathcal{I}(x_l) = \sum_{j=1}^{J} \mathbb{1}[v(j) = x_l]\Delta\mathcal{L} \tag{22}$$

where, x_l is the variable of interest, $v(j)$ the split variable at index j and $\Delta\mathcal{L}$ the difference in the loss function before and after the split on x_l.

Partial dependency

The concept of partial dependence relies on marginalising a variable and seeing what effect it has on the predictions. Let $S \subset \{1, 2, ..., p\}$ and C be the complement of S, further, let

$\boldsymbol{x}$ be our training data and $\boldsymbol{x_s}$ the coordinates in S of $\boldsymbol{x}$. The partial dependence function for a regression model is then defined by:

$$f_s = \mathbb{E}[f(\boldsymbol{x}_c, \boldsymbol{x}_s)] = \int f(\boldsymbol{x}_c, \boldsymbol{x}_s) dP(\boldsymbol{x}_c) \tag{23}$$

we estimate this equation by:

$$\hat{f}_s = \frac{1}{N} \sum_{i=1}^{N} \hat{f}(\boldsymbol{x}_{c_i}, \boldsymbol{x}_s) \tag{24}$$

where $\hat{f}$ is the statistical model we use and $\boldsymbol{x}_{c_i}$ the variable values used in training the model. [18] By plotting the partial dependence of the predicted variable and one of the independent variables, we can see more clearly what effect the individual predictive variable has on the response. This can be particularly interesting for an insurer as it is known in the insurance industry that some features have a strong influence on the frequency. For example, it is known in auto motive insurance that young drivers, especially males, tend to stand out in having a large amount of claims.

4 Case Study

This section focuses on building and evaluating the models covered in the background, using claims data from a Swedish insurer.

4.1 Data

The dataset used contains claims data for the all-risk cover of the company's housing insurance product, it consists of approximately 200,000 rows and covers feature variables such as, how large the insured property is, how many people live in property or the age of the insurance undertaker, but also the exposure time and number of claims, that constitute the frequency. The data set contains a very large number of rows where no claim has been made (97%) and therefore is quite imbalanced. This is partly dealt with

by the Poisson deviance, however it can still cause problems in modelling as the over represented zero-valued claims could easily be favoured by the model and cause a high accuracy without even considering the underrepresented data. We give an overview, in the form of histograms, of the number of claims and exposure times in Figure 4 and 5.

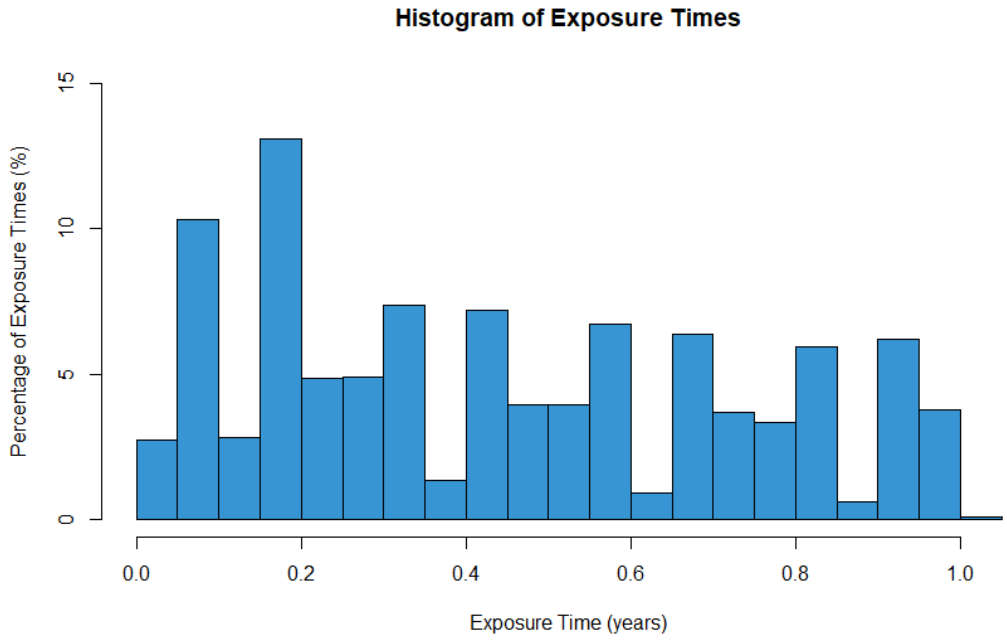

Figure 4: Histogram of Exposure

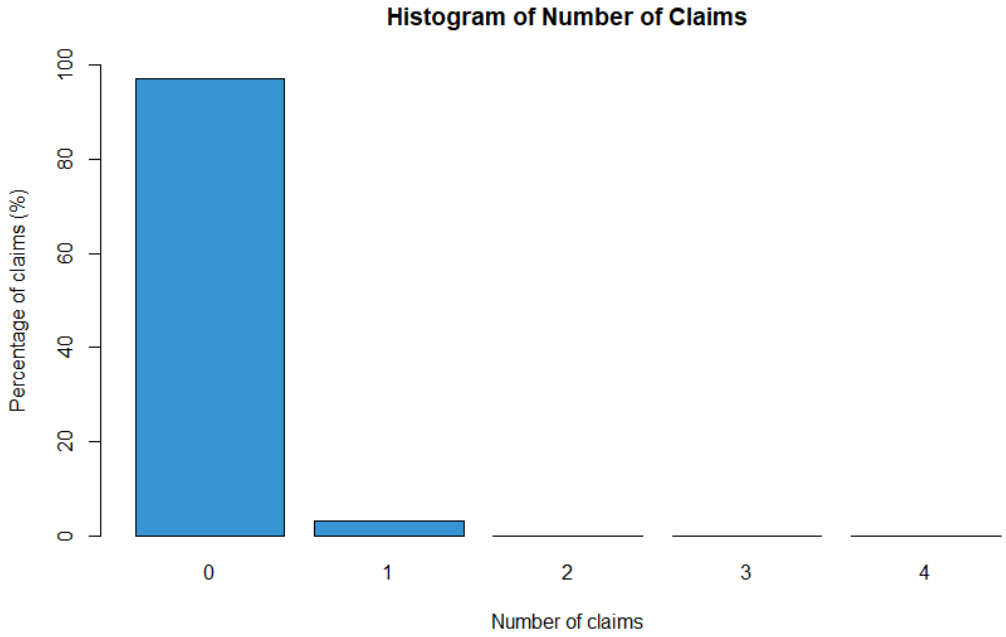

Figure 5: Histogram of Number of Claims

Before any modelling can be done the data needs to be prepared. In this case the data preparation focused on mainly three issues:

- Outliers
- Uncertain data
- Missing values

Outliers For the all-risk cover, only claims that have a severity of 50,000 SEK or less are covered by the insurance. Therefore, all values above 50,000 were set to 50,000. Moreover, there were several faulty observations, such as negative claims, which were omitted.

Uncertain data In paying a claim there are three important quantities, the amount incurred but not reported (IBNR), the amount reported but not settled (RBNS) and the amount that has been paid to the customer. The insurance company will reserve money to cover the RBNS and the IBNR, and after the claim has been made by the insured this reserve will start being paid out to the customer until the claim is closed (See Figure 6). There is a certain uncertainty in the RBNS, which could be an issue. Therefore the data was truncated so that the fraction of RBNS was relatively low.

Missing values Rows containing missing values where removed only if they were missing for an important feature or for one of the response variables, as most model implementations cannot deal with missing values. This of course reduces the amount of data, but given the size of the data set this is not a great loss. After the data preparation, around 140,000 observations were left.

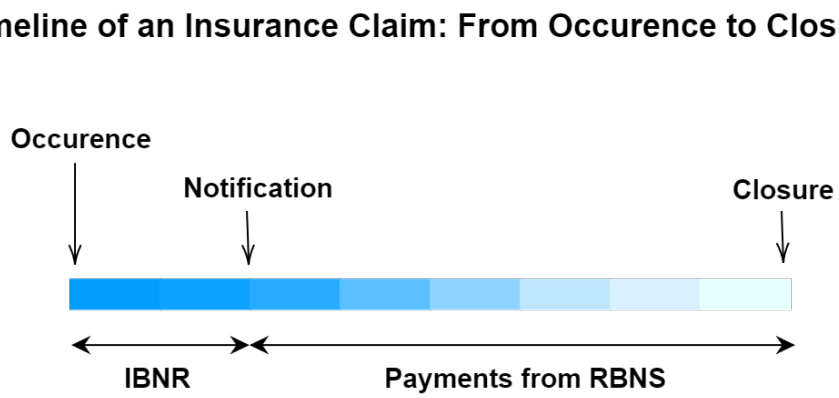

Figure 6: The insurance company pays the claim from their reserve until the closure time

4.2 Modelling

The first step of building a model is to choose what variables to use as features. This can be tricky as it is not always obvious in beforehand if there is a correlation between the response and a feature. Here, we chose to look at what variables were important in the GLM model currently used by the company, and assume these would also be important for the new models. The variables used in modelling are summarised in Table 1.

Table 1: Variable names of the dependent and independent variables used in the models

Variable name	**Description**
NO_CLAIM _NOT _NULL	No. claims
EXP_COV	Exposure time
AGE_INSUR_PERS	Age of the insured
NO_INSUR	No. people in the household
ACCOM_TYPE_NAME	Type of property
LIVE_AREA	Surface area

The models were implemented in R, primarily using an extension of the rpart package (found at `https://github.com/henckr/distRforest`) as well as the gbm package [19]. For supplemental R code please visit the following Github: `https://github.com/`

`samueltober/Insurance-ML`. For more information on implementation details and explanations of parameters see [19] and [20].

4.3 Results

In this section we present the results of the models covered in the background when used to predict frequencies using the insurance data set as training data. We evaluate the models based on three metrics: Variable importance, Poisson deviance on the 6 folds of the cross-validation and partial dependency plots.

4.3.1 Variable Importance

We evaluate the variable importance of the independent variables for each of the three models using the average of the optimal parameters found using the cross-validation scheme (See Table 2). Below are the results for each model.

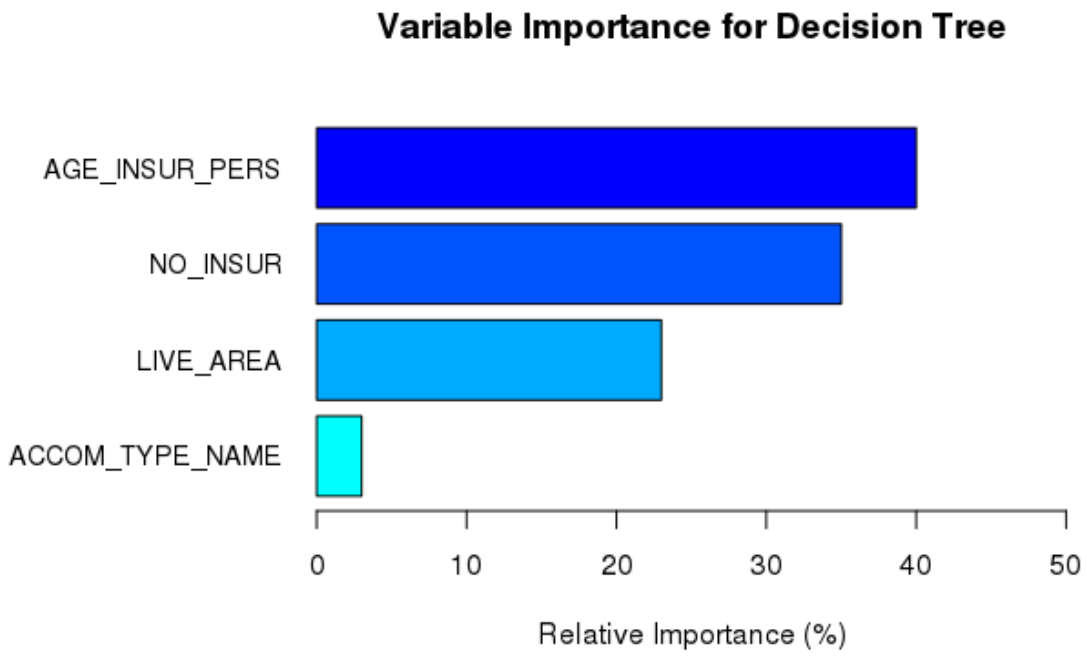

Figure 7: Relative variable importance in decision tree model

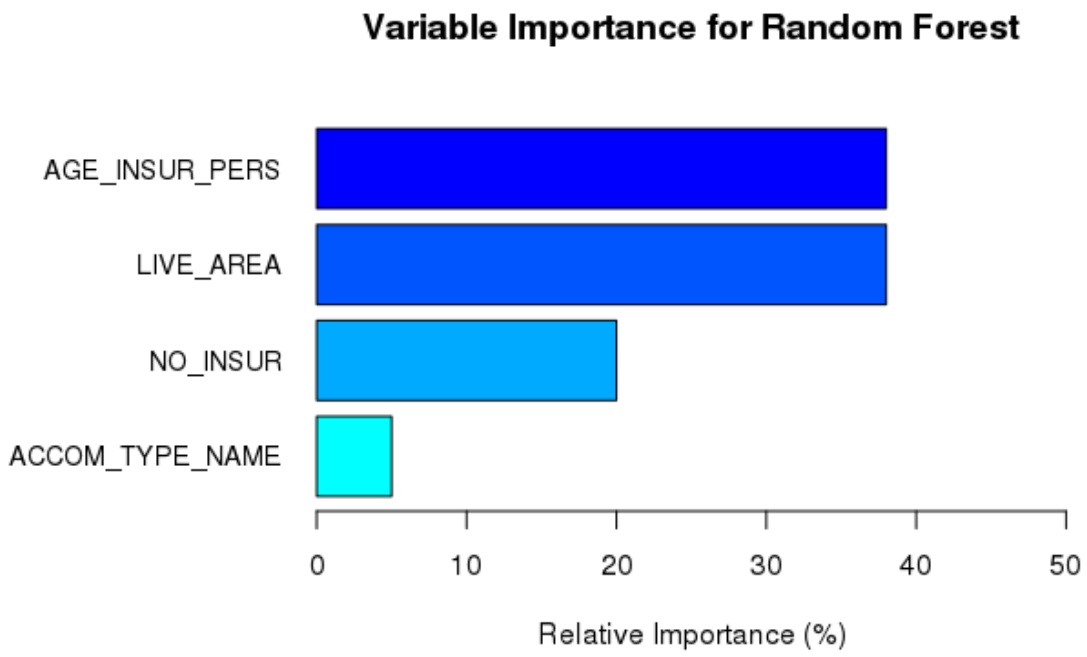

Figure 8: Relative variable importance in random forest model

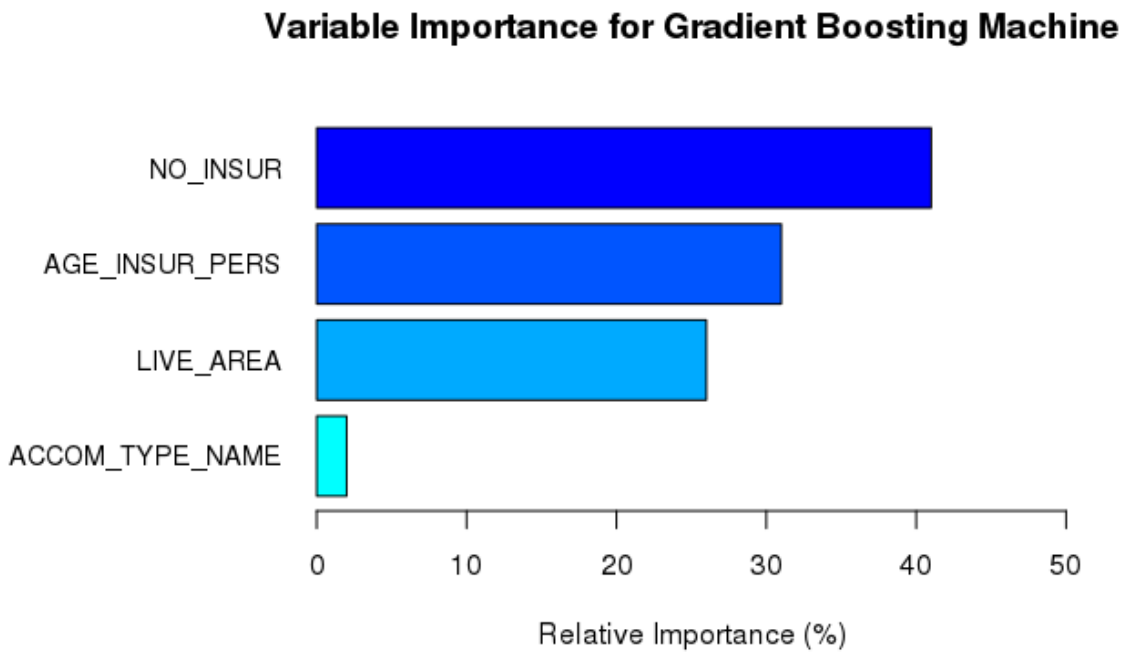

Figure 9: Relative variable importance in gradient boosting machine model

4.3.2 Cross-Validation

To choose the right value for the model parameters we implement the 6-fold cross validation scheme described in Section 3.2. This scheme was used for all three models, and the optimal parameters and Poisson deviances for the different folds are shown in Table 2 and Figure 10.

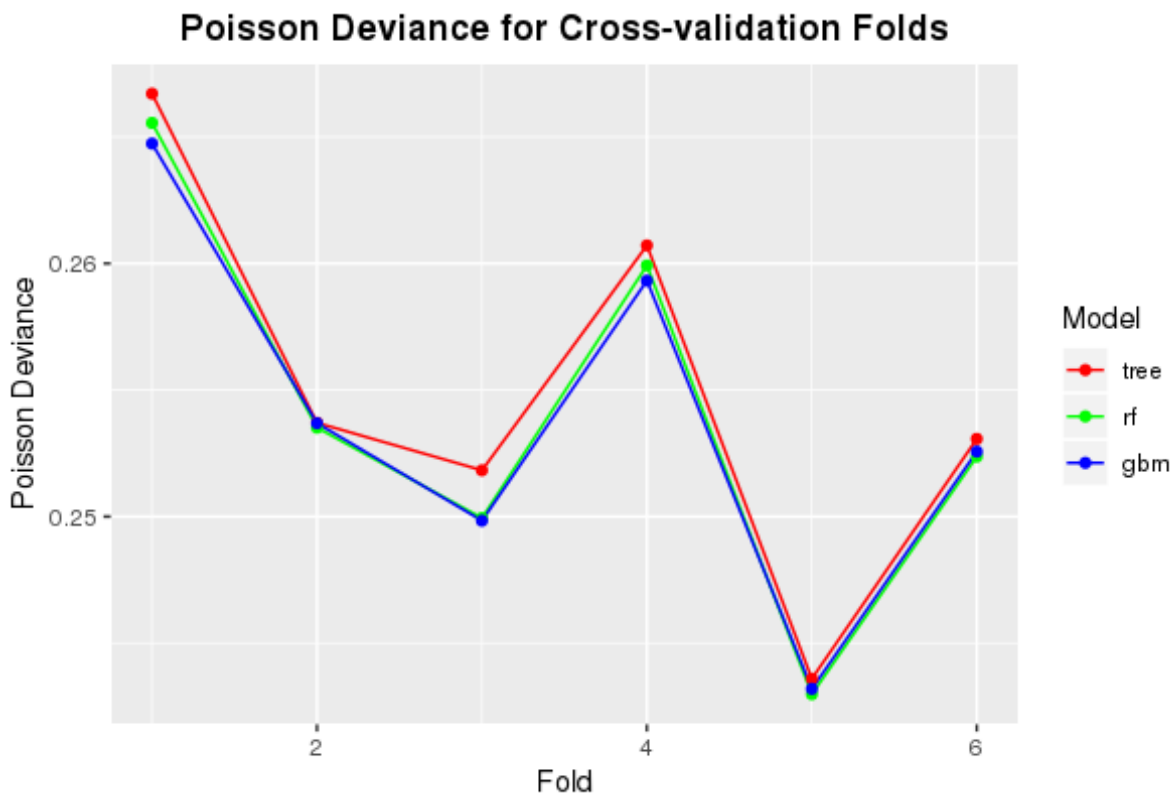

Figure 10: 6-fold cross-validation results for decision tree, random forest and gradient boosting machine

Table 2: Optimal parameters and average test deviance in cross-validation for decision tree, random forest and gradient boosting machine

Fold	**Decision Tree**	**Random Forest**		**GBM**	
	cp	No. trees	No. cand	No. trees	Interaction depth
1	0.00036	350	2	1400	5
2	0.00040	300	2	1100	5
3	0.00050	400	2	1200	4
4	0.00034	400	2	1800	3
5	0.00037	400	2	1000	4
6	0.00038	400	2	700	5
Avg. Deviance	0.255	0.254		0.253	

4.3.3 Partial Dependency Plots

Below are the results for the partial dependency plots, plotting the partial dependency for the age of the insured (Figure 11), the number of people in the household (Figure 12) and the surface area of the property (Figure 13) against the frequency. The partial dependency was calculated for each continuous variable, for all of the three models using the average of the optimal parameters in Table 2. In all plots we see that the lines become flat after a certain x-value, this is due to lack of data for these values.

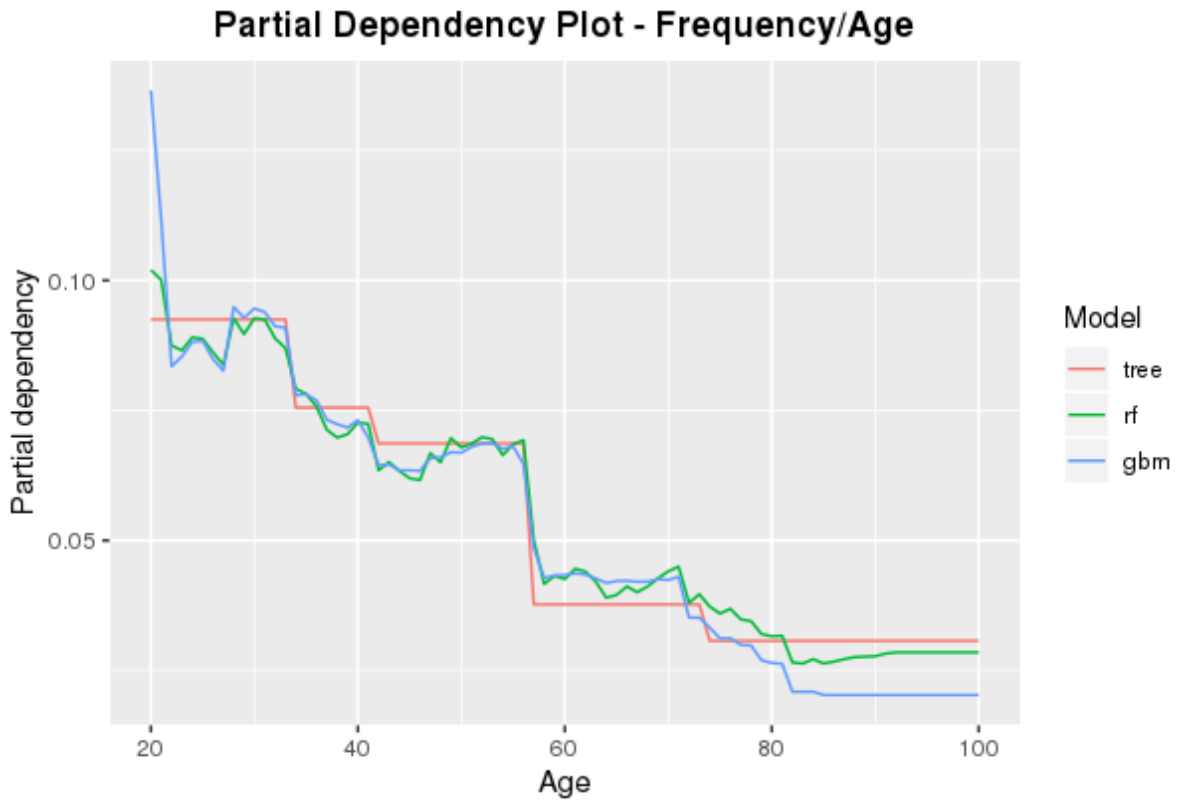

Figure 11: Partial dependency plot for age

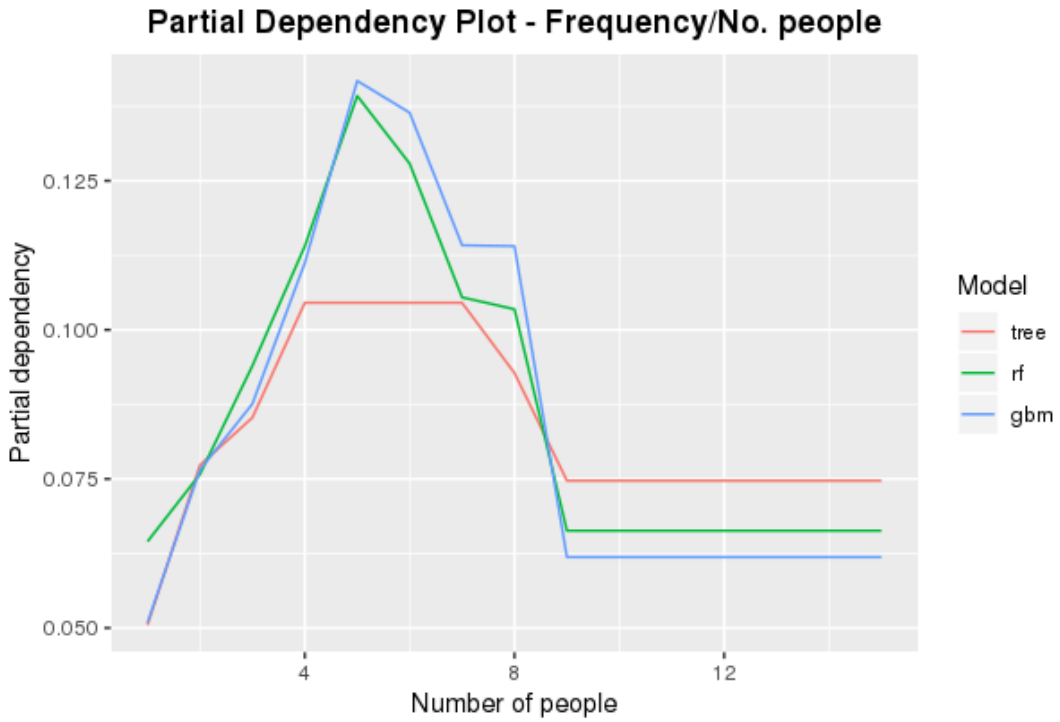

Figure 12: Partial dependency plot for the number of people in household

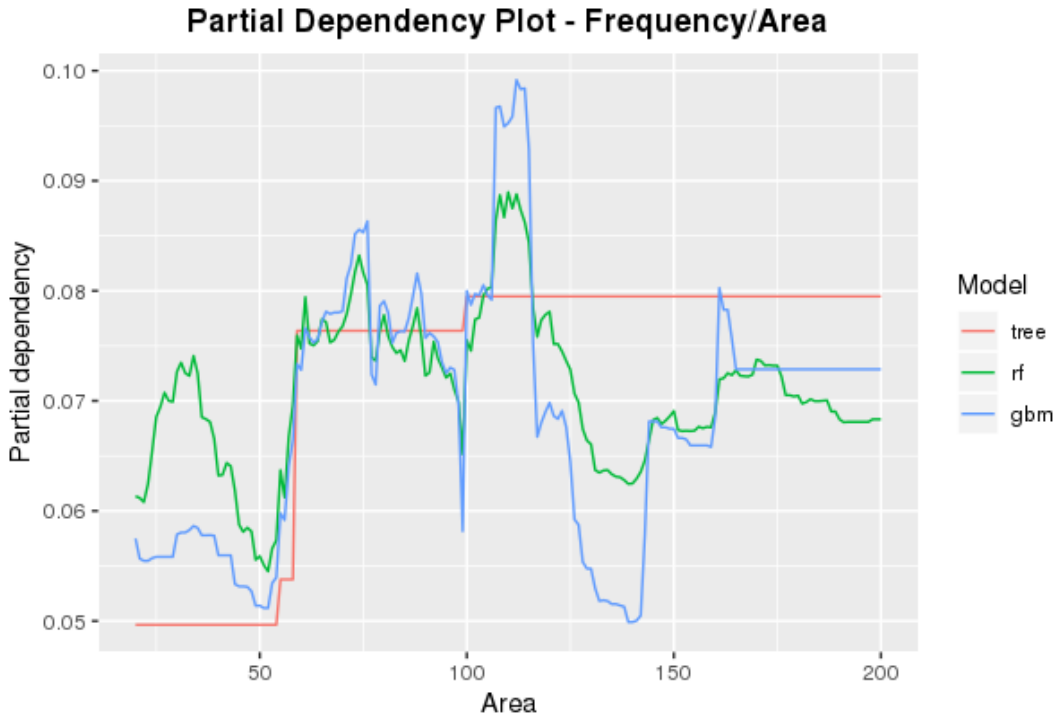

Figure 13: Partial dependency plot for surface area of the insured property

5 Discussions

Below the results presented in section 4 are discussed and conclusions based on these are drawn.

5.1 Variable importance

The variable importance plots for the three different models paint three quite different pictures. Nevertheless, the models still have some similarities. Firstly, neither model considers the accommodation type as an important feature, having the lowest relative importance for all three models. Secondly, all three models consider the age of the insured to be an important variable. Thirdly, regarding the number of people in the household, both the decision tree and the gradient boosting machine agree that this variable is important while the random forest favours it less. Lastly, the surface area is considered quite important for both the random forest and the gradient boosting machine, while the simple decision tree finds it less important. Variable importance is commonly used to select variables for modelling, however, as all variables in this case have an importance greater than zero, none of the variables should be discarded in modelling.

5.2 Partial dependency

We separately discuss the partial dependency for the three different continuous variables: Age, number of people in the household and surface area. In general we can note that the single decision tree gives a much coarse dependence, while both the gradient boosting machine and random forest are more smooth. The gradient boosting machine shows quite a large variance for some values, especially in Figure 13. Because the gradient boosting machine is an iterative process, it can be fairly sensitive to the data, which could explain the high variance. Nonetheless, all three models show signs of roughly the same trends.

Age of the Insured

We can see a general trend in that the number of claims filed seems to decrease with the age with a sudden drop in all three models around 60 years of age. The downward trend could be due to an increasing wariness as one gets older and/or the fact that younger people tend to take higher risks.

Number of People

For the number of people living in the household we see an increase in the number of claims with the number of people, with a peak around 5 people, and then a decrease. This decrease in claims, around 5 people could be explained by babies or children increasing the attentiveness of the parents and thereby reducing the number of filed claims.

Surface Area

The partial dependency plot for the surface area paints a rather volatile picture. All the models are in accord regarding the overall trend, with the exception of the gbm showing a rather high spike around 110 m^2 and drop around 140 m^2 compared to the other models. One could speculate in explanation for trends shown in this plot but generally there is not an obvious trend as in Figure 11 and Figure 12.

5.3 Cross-Validation Performance

The cross-validation results show that the gradient boosting machine on average has the best predicitve performance, followed by the random forest and the simple decision tree performs the worst. In comparison, Henckaerts et al. [8] found that the gradient boosting machine had a significantly better performance compared to the random forest and decision tree. The reason we do not see this clearly in our results could be due to the data quality, the tuning grid used in cross-validation or other parameter settings in the gradient boosting machine implementation. Due to limited computing power, the cross-

validation tuning was not carried out as extensively as one could have wished for, which could mean that the models were not perfectly tuned. Although the simple decision tree has worse performance than the other two models, it has the advantage of being simple and very fast to train. Depending on the application, it could be favourable to use a more simple and fast model so we cannot rule out the simple decision tree. However, in a risk prediction context, a better predictive performance would be a priority since it could potentially increase the margins of the insurance company.

6 Further Research

The scope of this study is clearly limited, and there is a great deal yet to investigate. First and foremost, this study solely focused on predicting the frequency of claims and did not pay any attention to the severity. Severity is equally important in assessing the risk of a customer and must therefore be taken into account. Secondly, as mentioned previously there are usually interactions at play between the independent variables. We did not delve into this in this paper, however these effects can be of great importance in rate making as they can give a deeper understanding of certain variable combinations which in the end could influence the pricing strategy. Thirdly, as mentioned above, due to lack of computing power, the cross-validation scheme was perhaps not thorough enough to give a significant result. Hence, a more elaborate scheme should be employed in any further research. Finally, the models were not compared to a state-of-the-art GLM model, which in the end is necessary to draw any conclusions on whether the ML models have potential contribute to a more profitable pricing strategy compared to industry standards.

References

[1] Werner et al., *Basic Ratemaking*, May 2016, Fifth edition. Casualy Actuarial Society

[2] Konsumenternas.se. Om hemförsäkring, 2015. https://www.konsumenternas.se/forsakring/boende/om-hemforsakringar

[3] M. Denuit and S. Lang. Non-life rate-making with Bayesian GAMs. Insurance: Mathematics and Economics, 35(3):627–647, 2004

[4] Embrechts, Paul, (1998), S.A. Klugman, H.H. Panjer and G.E. Willmot (1998): Loss Models: From Data to Decisions. Wiley, New York, ASTIN Bulletin, 28, issue 1, p. 163-164, https://EconPapers.repec.org/RePEc:cup:astinb:v:28:y:1998:i:01:p:163-164\textunderscore01.

[5] Wuthrich, Mario V. and Buser, Christoph, Data Analytics for Non-Life Insurance Pricing (June 4, 2019). Swiss Finance Institute Research Paper No. 16-68. Available at SSRN: https://ssrn.com/abstract=2870308orhttp://dx.doi.org/10.2139/ssrn.2870308

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

[7] Gelman, A., & Hill, J. (2006). Data Analysis Using Regression and Multilevel/Hierarchical Models (Analytical Methods for Social Research). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511790942

[8] Henckaerts, Roel Côté, Marie-Pier Antonio, Katrien Verbelen, Roel. (2019). Boosting insights in insurance tariff plans with tree-based machine learning.

[9] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. Classification and Regression Trees. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, New York, 1984.

[10] Jason M. Klusowski *Analyzing CART*, 2019 https://arxiv.org/pdf/1906.10086.pdf

[11] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York :Springer, 2013. p.33-42

[12] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York :Springer, 2013. p.309

[13] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York :Springer, 2013. p.315-316

[14] Breiman, L. Bagging Predictors. Machine Learning 24, 123–140 (1996). https://doi.org/10.1023/A:1018054314350

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

[16] Friedman, Jerome. (2000). Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics. 29. 10.1214/aos/1013203451.

[17] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)

[18] Pearson, R. Interpreting Predictive Models Using Partial Dependence Plots. CRAN (Feb. 2020). https://cran.r-project.org/web/packages/datarobot/vignettes/PartialDependence.html

[19] B. Greenwell et al. Package "gbm" (Jan. 2019) https://cran.r-project.org/web/packages/gbm/gbm.pdf

[20] T. Therneau & Beth Atkinson. Package "rpart" (Apr. 2019) https://cran.r-project.org/web/packages/rpart/rpart.pdf