Norwegian School of Economics
Bergen, Spring 2020

NHH

Predictive modelling of customer claims across multiple insurance policies

An empirical study of how individual customer insurance data can be used to assess customer risk across multiple insurance products by employing machine learning and advanced ensemble techniques

David Høysæter and Endre Larsplass

Supervisor: Geir Drage Berentsen

Master's thesis in Business Analytics
MSc in Economics & Business Administration

NORWEGIAN SCHOOL OF ECONOMICS

Contents

Abstract

In this master thesis, we have analysed how individual insurance customer data can be used to assess customer risk across multiple insurance policies. Our dataset contains 63 variables about the characteristics of each customer and five associated response variables provided by Frende Forsikring. We have modelled the responses for claim propensity, claim frequency, and total claim size for each customer. To evaluate the value of this customer data, we have used multiple machine learning algorithms. These include XGBoost, LightGBM, random forest, GLM and deep neural networks. We have also used different ensemble techniques to gain further performance improvements from these models.

By comparing results achieved using customer insurance premium as the only explanatory variable to the results achieved using all the additional customer characteristics we could observe a considerable increase in predictive performance. Our findings show that gradient boosting techniques can increase performance compared to generalized linear models. We also observed that using multiple models in ensembles can increase performance compared to any single model when assessing customer claim propensity and frequency. Although we found stacked ensembles using multiple underlying models to provide increased performance when used on claim propensity and frequency, we found a strong case for the use of generalized linear models when modelling total claim size. Our thesis proposes a novel three-step ensemble model that uses claim propensity and claim frequency to determine the total claim size of a customer, which may improve performance of total claim predictions.

Overall, our results show promise in using individual customer data to supplement the traditional individual policy risk assessments. The results also underline the potential of advanced ensembles to increase predictive performance on the individual customer data. The results accentuate the importance of selecting the appropriate models and suitable error metrics to achieve good predictive performance across different response variables. Our findings illustrate the transparency issues associated with using highly flexible statistical learning tools when compared to generalized linear models.

Acknowledgements

We want to express our gratitude towards our supervisor Geir Drage Berentsen for his insights, collaboration, and engagement throughout the process of writing our master thesis. Furthermore, we would like to thank Frende Forsikring for providing us with the opportunity and necessary data for our research on how customer data can provide value through individual customer risk modelling within casualty insurance. In particular, we want to thank actuary Eivind Herfindal Reikerås and business analyst Anders Dræge at Frende Forsikring for a helping hand and for providing us with valuable insights.

2 Introduction

2.1 Background

This master thesis is written to conclude our master's degree at Norwegian School of Economics (NHH).

Differentiated pricing in casualty insurance is widespread both within insurance companies and academia. Estimating the total expense of each customer per year, per insurance policy, is integral to the profitability and sustainability of an insurance company. With increasing amounts of data available on each customer, and customers often holding multiple insurance products, it has become increasingly viable to model aggregated customer risk in addition to modelling individual policy risk. These models are computationally expensive, but have become increasingly viable due to an exponential increase in computing power and new statistical models allowing greater flexibility and the use of many explanatory variables.

2.2 Motivation

The insurance industry in both Norway and internationally is characterized by fierce competition for customers. Although there are slight differences between the insurance companies and their product offerings, the policyholder will perceive products to be largely similar. The homogenous nature of insurance products means pricing is often the primary criterion for the customer to base their choice of insurance company on. We can assume that less risky customers are willing to pay less for their insurance than risky customers, which makes it essential to be able to offer less risky customers competitive prices. If the prices are set too high, the only customers susceptible to take out insurance are risky customers. This is known as an adverse selection issue, where the incentives of the two parties are misaligned and there is an asymmetric information pattern where the customer knows more about its own behaviour than the insurance company does. Pricing customer risk correctly is therefore critical to attract and keep profitable insurance customers and to stay competitive.

Today, Frende have well-performing models to calculate the risk of individual policies. With increasingly more data available on customer and customer relationship level, they are interested in evaluating individual customer risk across one or multiple individual policies in addition to the existing individual policy risk models. Frende provides a broad range of

different private customer insurance products, and we want to explore whether individual customer information associated with all these products can provide additional value to their existing risk assessments. We will do this by modelling the probabilities of each customer making a claim, the frequency of claims, and the total claim amount in a given year, and see if predictions benefit from the additional customer data provided. To achieve this, we will use different machine learning models in multiple configurations to see how we can best leverage the individual customer data across these three response variables. We can describe the total claim as

$$U_i = \sum_{k=0}^{A_i} S_{i,k} B_i \quad where\ B_i \rightarrow \{0,1\}$$

where U_i is the total claim U for customer i, A_i is the number of claims A for customer i, and $S_{i,k}$ is the average claim size given by customer i and claim k, and B_i a binary indicator B describing if customer i has made a claim.

2.3 Utilization of R and R packages

We are performing all modelling and calculations in our thesis using the open-source programming language R. R is free to use and provides an extensive library of statistical extensions through packages available in CRAN (The Comprehensive R Archive Network). We have extensively used the "caret" package to provide a uniform interface for classification and regression models. Our neural network models use the R-package "Keras", which employs the underlying open-source machine learning platform Tensorflow. The individual packages used for our models will be described in the method section.

3 Method

3.1 Machine learning in statistical learning

Breiman (2001) distinguishes between two statistical modelling paradigms. One assumes data are generated by a given stochastic data model, while the other uses algorithmic models and treats the data mechanisms as unknown. Machine learning is considered part of the latter and focuses on best predicting the dependent variable and puts less emphasis on the relationship between the dependent variable and the predictors.

3.1.1 Supervised and unsupervised learning

The learning process of machine learning algorithms can be divided into two main categories, supervised and unsupervised learning (Hastie et al., 2004). Supervised learning algorithms build mathematical models from data sets containing both inputs and the desired output (Russel & Norvig, 2013). Unsupervised learning is algorithms that learn from non-labelled data and tries to find commonalities between observations and react to new data based on these findings. In our thesis, we will focus on supervised learning, as all our observations are labelled with response variables. The response variables can take the form of a continuous response, a binary classification response with two potential outcomes, or a multi-classification response with multiple potential classes.

3.1.2 Training and test data

For training purposes, datasets are often divided into multiple parts. Training sets are used to fit the optimal parameters to minimize a pre-determined loss function. To benchmark the model, part of the dataset is withheld to see how well the models can predict these unseen observations. The withheld data is called the test set. Some algorithms provide the opportunity to tune hyperparameters to improve model performance on a given dataset. A hyperparameter is a parameter whose value is set before the model training begins and is used to maximize the usefulness of the learning approach (Claesen & De Moor, 2015). In such instances, the training set may be divided into two parts: One to train the model on, and one validation set to evaluate model performance on unseen data. The validation set provides an unbiased evaluation of the model performance on unseen data when training, and is used to prevent overfitting (Ripley, 2007). Overfitting is when the model adapts well to the dataset it is trained on, but it does not generalize well on unseen data.

3.1.3 Cross-validation and hyperparameter tuning

One downside of using validation sets to evaluate model performance outside the training set is that we essentially "*waste*" a part of the training set, as those observations cannot be used to train the model. One way to use all observations for training purposes, but still make sure the model does not overfit, is by using cross-validation. Cross-validation has several benefits over a dedicated validation set. A common technique is k-fold cross-validation (McLachlan, Do & Ambroise, 2005). It partitions the training set into *k* equal sized subsamples, retains a single subsample as validation data to test the model, and the remaining $k-1$ subsamples are used as training data. The process is repeated *k* times with each of *k* subsamples used exactly once as the validation data (Breiman & Spector, 1992). The results can then be averaged for comparison purposes.

The partitions of data can be selected to make sure the mean response value is approximately equal by using stratified k-fold cross-validation, especially useful in training classification models (Molinaro, Simon, & Pfeiffer, 2005). In stratified k-fold cross-validation, the partitions are selected so that the mean response value is approximately equal in all the partitions. In the case of binary classification, this means that each partition contains roughly the same proportion of the two types of responses.

3.1.4 Bias-variance tradeoff

When choosing the appropriate models to predict response variables, one important consideration is the bias-variance tradeoff. We can illustrate the bias-variance tradeoff by decomposing the expected mean squared error (MSE) into two fundamental quantities in the following way

$$E\left(y_0 - \hat{\mathrm{f}}(\mathrm{x}_0)\right)^2 = Var\left(\hat{\mathrm{f}}(x_0)\right) + \left[Bias\left(\hat{\mathrm{f}}(x_0)\right)\right]^2.$$

Here, y_0 is the observed response, and $\hat{f}(x_0)$ is the response function of explanatory variables x_0. We see that the MSE ($E\left(y_0 - \hat{\mathrm{f}}(\mathrm{x}_0)\right)^2$) can be decomposed into a variance term $Var\left(\hat{\mathrm{f}}(x_0)\right)$ and a bias term $\left[Bias\left(\hat{\mathrm{f}}(x_0)\right)\right]^2$. To minimize the MSE, a statistical model needs to achieve low bias and low variance simultaneously. Variance refers to the amount by which $\hat{f}$ would change if we used a given model on a different training data set. Bias refers to the

error introduced by approximating a complex real-life problem by a simpler model. A linear regression model would have a high bias if modelling a non-linear problem, and a low bias if estimating a linear problem. Using flexible methods generally introduce more variance and decrease bias, while less flexible methods have lower variance and increased bias (James et al., 2013). Finding an optimal tradeoff between the two extremes is key to increase model performance. This is called the bias-variance tradeoff, and it is an important consideration when finding the appropriate model to a given problem. Increasing model flexibility will decrease MSE on the training data but the improved results might not transfer to the test set. Instances where training performance increases but test performance on out-of-sample data decreases is called overfitting. Using validation and cross-validation sets is critical to ensure models are not overfitted.

3.2 Probability distributions

A probability distribution is a mathematical function that describes the probabilities of the occurrence of an experiment's possible outcomes (Ash, 2008). In the following section, we will introduce the probability distributions that are used later in this thesis. We will present conventions for how each of the distributions can be parameterized and describe the parameters.

3.2.1 Normal distribution

The normal distribution, also known as gaussian distribution, is the best-known probability distribution. It is often called a bell curve and is applied frequently as an analytical tool in statistics. The gaussian distribution is a continuous probability distribution with support on $Y = \in (-\infty, \infty)$. The probability density function (PDF) of the normal distribution is perfectly symmetric and light-tailed. It can be formulated in the following way

$$f(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2}$$

where μ is the mean, which is equal to the mode and median in a normal distribution, σ is the standard deviation and σ^2 is the variance. We say that the response *Y* follows a standard normal distribution when $\mu = 0$ and $\sigma^2 = 1$.

3.2.2 Gamma distribution

There are three different parameterizations of the gamma distribution that are commonly used. We will focus on the exponential distribution with shape parameter α and an inverse scale parameter $\beta = \frac{1}{\theta}$, where β is the rate parameter, and θ is the scale parameter. The size of the shape parameter α affects the skewness and the kurtosis level of the distribution. The gamma distribution is a continuous probability distribution with support on $Y = \in (0, \infty)$. The probability density function (PDF) of the gamma distribution is moderately skewed and moderately heavy-tailed. It can be parameterized the following way

$$Y \sim \Gamma(\alpha, \beta) \equiv \text{Gamma}(\alpha, \beta)$$

$$f(y; \alpha, \beta) = \frac{\beta^{\alpha} y^{\alpha-1} e^{-\beta y}}{\Gamma(\alpha)} \; for \; y > 0 \;\; \alpha, \beta > 0$$

where the gamma function is

$$\Gamma(\alpha) = (\alpha - 1)!, for\ all\ positive\ integers$$

The expectation or mean of the distribution can be written as $\frac{\alpha}{\beta}$, and the variance of the distribution can be written as $\frac{\alpha}{\beta^2}$. The gamma distribution is often used to model claim sizes in insurance applications.

3.2.3 Log-normal distribution

Log-normal is a continuous probability distribution with support on $Y = \in (O, +\infty)$. If $log(Y)$ follows a normal distribution with expectation μ and variance σ^2, we say that Y follows a log-normal distribution with the parameters μ and σ^2. The expectation or mean μ explains differences in expectation and variance. The PDF of the log-normal distribution can be described as

$$log(Y) \sim N(\mu, \sigma^2)$$

$$f(y; \mu, \sigma^2) = \frac{1}{y} * \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{(lny-\mu)^2}{2\sigma^2}\right)}$$

where expectation or mean is given by

$$E(Y) = \exp\left(\mu + \frac{\sigma^2}{2}\right)$$

and the variance is

$Var(Y) = [exp(\sigma^2) - 1]\, exp(2\mu + \sigma^2)$.

The log-normal distribution is frequently used to model claim sizes in insurance applications, just like the gamma distribution.

3.2.4 Inverse gaussian distribution

The inverse gaussian distribution is a continuous probability distribution with support on $Y = \in (0, \infty)$. The distribution is very skewed with a steep top. The probability density function (PDF) of the Inverse Gaussian distribution can be written the following way

$$f(y; \mu, \lambda) = \sqrt{\frac{\lambda}{2\pi y^3}}\, e^{\left(-\frac{\lambda(y-\mu)^2}{2\mu^2 y}\right)}, for\ y > 0, \mu > 0, \lambda > 0$$

where expectation or mean is given by

$$E(Y) = \mu$$

and the variance is

$$\mathrm{Var}(Y) = \frac{\mathrm{m}^3}{\lambda}$$

The inverse gaussian distribution has positive support and is therefore particularly useful in insurance and economic data applications where non-positive responses do not occur.

3.2.5 Poisson distribution

The Poisson distribution is a classical probability distribution describing count data. The probability distribution is discrete with support on $Y = \in \mathbb{N}_0$ (all natural numbers starting from 0). The Poisson distribution only has one parameter μ that defines the expectation and the shape of the probability mass function (PMF). The PMF can be written the following way

$$f(y; \lambda) = Pr(Y = y) = \frac{\lambda^y e^{-\lambda}}{y!}$$

Where the mean equals the variance

$$E(Y) = \mathrm{Var}(Y) = \lambda$$

The Poisson distribution is often used to describe count data, and in insurance applications it is commonly used to describe the claim frequency.

3.2.6 Negative binomial distribution

The negative binomial distribution is a discrete distribution with support on the integer number of successes $Y = \in \mathbb{N}_0$. It is similar to the Poisson distribution in many aspects, but it allows for overdispersion, which occurs when the variance is larger than the expectation. The probability density function (PDF) can be formulated as

$$f(y; r, p) \equiv P_r(Y = y) = \left(\frac{y + r - 1}{y}\right) p^r (1 - p)^y$$

where *r* is the number of successes, *y* is the number of failures, and *p* is the probability of successes.

The expectation or mean is given by

$$E(Y) = \mu = \frac{pr}{1 - p}$$

and the variance is given by

$$\mathrm{Var}(Y) = \mu(1 + y\mu)$$

The negative binomial distribution is useful for count data when the data is overdispersed, as overdispersed data makes the Poisson distribution less suitable.

3.3 Models

We have used three main types of models in our thesis to predict claim risk, claim frequency, and total claim size of customers. We will distinguish these as generalized linear models, decision trees, and neural networks.

3.3.1 Generalized linear model

Nelder and Wedderburn (1972) introduced Generalized Linear Model (GLM). It is a flexible generalization that allows for response variables with error distribution models other than a normal distribution. The GLM consists of three elements: an exponential family of probability distributions, a linear predictor which is a linear function of covariates, and a link function, which is a function of the response variable's mean, and equal to the linear predictor (Pan & Yang, 2011).

The GLM framework uses the response variable Y_i, an independent stochastic variable with distribution f having mean $\mu_i = E(Y_i \mid X_i = x_i)$ which depends on explanatory variables x_i through link function g, so that $g(\mu_i) = \eta_i$, where $\eta_i = x_i^T \beta$ is called the linear predictor. The linear predictor η_i has linear coefficients $\beta = \{\beta_0, \ldots, \beta_p\}$ and the distribution of the response variable is in the exponential family. Parameter estimation in GLM is done via maximum likelihood estimates using iteratively reweighted least squares (IRLS) or Newton-Raphson (N-R). Solving weighted least squares can be described as a minimization problem of the form

$$\beta^{(t+1)} = \underset{\beta}{\arg\min} \sum_{i=1}^{n} w_i(\beta^{(t)}) \, |y_i - f_i(\beta)|^2$$

where β are the parameters which minimize the linear regression problem, and w_i are the weights updated after each iteration for $\beta^{(t)}$ by minimizing error term $y_i - f_i(\beta)$.

There are several reasons why GLMs are suitable for actuarial modelling. The methodology allows the user to choose the distribution and link function based upon knowledge of the response distribution, and there are numerous software packages that can be used for estimation purposes. In practice, it is also easy to interpret how different explanatory variables impact the response variables, especially compared to more sophisticated machine learning methods.

3.3.2 Decision tree

A decision tree is a well-known model used in many different applications. It is highly interpretable, and it is easy to visualize the reasoning behind its response. In machine learning, it can be used to solve both classification and regression problems. To make decision trees efficient to solve statistical problems, there are two important considerations: how decision trees determine its splits, and the shape of the overall decision trees. Decision tree algorithms use nodes to represent explanatory variables, branches to represent decisions, and leaf nodes to represent responses.

Decision trees are usually constructed top-down by choosing the variable at each step that splits the best set of items (Rokach & Maimon, 2005). Different algorithms use different metrics for measuring what the best item is. To determine the optimal split in decision trees, it needs to evaluate the impurity of the sample it evaluates. The impurity is decided by how homogenous the sample is. A homogenous sample will be considered pure, while heterogeneous samples are impure as there is more variation across the population. A common way to find the impurity is by using the Gini index. It is a measure of inequality in the sample, represented by a value between 0 and 1. A Gini index value of 0 means the sample is perfectly homogeneous, while a Gini index value of 1 indicates inequality or heterogeneity among its observations. The Gini impurity for a set of items with J classes and p_i fraction of items labelled with class i in the set can be formulated as

$$Gini\ impurity = 1 - \sum_{i=1}^{J} p_i^2 .$$

Classification and regression trees (CART) is an umbrella term used to refer to classification and regression decision trees introduced by Breiman et al. (1984). It uses the Gini impurity for classification splits and the relevant improvement in the sum of squared errors between the node and its child nodes after the split in regression.

3.3.3 Neural network

Neural networks are computing systems vaguely inspired by biological neural networks. They are based on a collection of connected units or nodes, called artificial neurons, which resemble neurons in a biological brain (Chen et al., 2019). For a single node, there is a set of

observed explanatory variables X_i, and a response variable Y. A network with multiple hidden layers is called a deep neural network (Schmidhuber, 2015). A deep learning network consists of an input layer, hidden layers, and an output layer. The hidden layer can be formulated as

$$N_j^1 = f(b_j^0 + \sum_{i=1}^{I} w_{ji}^0 N_j^1)$$

and the output layer as

$$Y = g(b^1 + \sum_{j=1}^{J} w_j^1 N_j^1)$$

where the hidden layer consists of intermediate nodes N_j^1. The node takes several inputs, x_1, …, x_n, and each individual node has an internal set of weights b, w_1, …, w_n, and an activation function f. The inputs connect to the intermediate nodes, and the intermediate nodes connect to the outputs. Each layer consists of neurons that take inputs and transform them into representations useful for solving the given problem (Borovykh, Bohte, & Oosterlee, 2017). These representations are non-linear and use an activation function to pass on to the next layer until the output layer is reached (Mueller & Massaron, 2016). By minimizing a given loss function, the neural network learns its optimal parameters (Goodfellow et al., 2016). These features allow neural networks to model complex non-linear relationships.

Deep neural networks are inherently exposed to vanishing gradients. This is a problem in which multiple layers lead to products of gradients, and the gradient becomes very small so that the product vanishes. The opposite problem of exploding gradients, where the gradients become too large, could also cause the algorithm to return unsatisfactory results. This problem is alleviated by using normalization to stop gradients from exploding or vanishing (Pascanu, Mikolov, & Bengio, 2013).

Using deep neural networks have some advantageous features. The hidden layer acts as higher-level features of the data, and output layer weighs the features to make the final prediction, which makes deep neural networks an automated feature engineering algorithm. This reduces the need for manual feature engineering, which is time-consuming and often dependent on domain knowledge. It has also been discovered how multiple hidden layers create an informational bottleneck leading the deep neural network to zero in on the correct classification (Tishby & Zaslavsky, 2015).

3.4 Ensemble techniques

Several methods can be used to improve the model accuracy of the fundamental models described in section 3.3. An effective method is to use ensembles of more than one model. We will describe the advantages associated with ensembles and what methods we have applied in our thesis in this section.

3.4.1 Simple ensemble methods

There are several reasons to use ensemble-based systems when predicting an outcome. A set of models with similar training performances may have significantly different generalization performance. Even when generalization performance is similar, different models often perform differently when the generalization performance is not sufficiently representative of the future data samples. If we had a model with perfect generalization performance, there would be no need to resort to ensemble techniques. In reality, noise, outliers, and overlapping data distributions make such a model an impossibility. We can assume that individual models make errors in slightly different instances. If each model makes different errors, then a strategic combination of these models can reduce the total error (Polikar, 2006).

The intuition of ensembles is similar to that of doctors consulting each other to arrive at the correct diagnosis. It is especially useful when using sophisticated modelling tools such as neural networks, which are prone to overfitting. Such models may perform well in most cases but make large mistakes in others. Combining several models makes sure that the predictions are not heavily dependent on any singular model. It can also be useful to handle particularly large volumes of data in instances model performance is a bottleneck, and complex problems where several models are needed as the problem is too advanced for any single model to solve (Polikar, 2006). In this section, we will look at a few simple techniques: majority voting, averaging, and weighted averaging.

3.4.1.1 Majority voting

Majority voting is a method used for ensemble classification problems. There are three versions of majority voting. Unanimous voting in which all three models, or classifiers, agree on the outcome, simple majority where at least half of classifiers agree on the outcome, or plurality voting in which the outcome which has most the most votes are selected. Unanimous voting and simple majority can be especially useful in instances where incorrect predictions

are considered more costly than correct predictions, where unanimous voting is the most extreme measure to avoid incorrect predictions. We can define the decision of plurality, from this point on referenced to as majority voting, from the t^{th} model with the following formula

$$\sum_{t=1}^{T} d_{t,j} = \underset{j=1}{\overset{C}{max}} \sum_{t=1}^{T} d_{t,j}$$

where $d_{t,j} \in \{0, 1\}$, $t = 1... ,T$ and $j = 1, ...C$, where T is the number of classifiers and C is the number of classes. If the t^{th} classifier chooses class ω_j, then $d_{t,j}$ = 1, and 0 otherwise. According to the Condorcet Jury Theorem (Boland, 1989) regarding audience polling, if each audience member has a higher probability than ½ of giving the correct answer, a large enough audience should approach a probability of success of 1. These principles have also been found to be transferrable to majority voting (Kuncheva, 2005).

3.4.1.2 Averaging

Averaging is a simple algebraic non-trainable combiner of continuous outputs. It can be used to make predictions in regression problems or calculating probabilities in classification problems. It can be expressed with the following formula

$$\mu_j(x) = \frac{1}{T}\sum_{t=1}^{T} d_{t,j}(x)$$

where μ_j is the average of j^{th} model outputs within a normalization factor $1/T$, and T is the number of models used.

3.4.1.3 Weighted averaging

Weighted averaging is an extension of the averaging method. In this method pre-determined weights are used to define the importance of each model in the final prediction. It can qualify as both a trainable and non-trainable combination rule, depending on how the weights are obtained (Polikar, 2006). We can express weighted averaging using T weights, w_1,..., w_T *as*

$$\mu_j\ (x)\ =\ \sum_{t=1}^{T} w_t d_{t,j}(x)$$

An example of how a trainable weighted average can be applied, is by using cross-validation to determine the performance of several models on training data. The model weights can be determined by minimizing the loss function based on the predicted responses. These weights are then carried to the out-of-sample predictions. It can also be non-trainable, based on intuition and experience on how well different models tend to perform on similar data.

3.4.2 Bagging

In addition to simpler ensemble methods like majority voting and weighted averaging, there are more complex approaches to create an ensemble of models. In our thesis, we will divide these into four categories: stacking, blending, bagging, and boosting. The purpose of these ensembles is the same as the simpler ensemble methods shown in the previous section, but the approaches are different. We will use stacking, bagging, and boosting to enhance our model performance.

Bagging predictors is a method for generating multiple predictors and using these to get an aggregated predictor. When predicting continuous outcomes, the aggregation averages over the versions, and in classification problems, it does a plurality vote. It uses multiple training subsets formed by bootstrapping the data set and uses these as new learning sets to create the individual predictors. This method can give substantial gains in accuracy in both classification and linear regression problems. The instability of the prediction method employed is vital in its usefulness. If small changes made to the training data easily results in changes of the predictor constructed, then bagging can improve accuracy. The principles of bagging can be illustrated using the following pseudocode.

Given a training set $X = x_1, ..., x_n$ with responses $Y = y_1, ..., y_n$, bagging B times selects a random sample with replacement of the training set and fits trees to these samples:

For $b = 1, ..., B$:

1. Sample, with replacement, n training examples from $X, Y; X_b, Y_b$.
2. Train a classification or regression tree f_b on X_b, Y_b.

When the model is trained, we can see how predictions on out-of-sample observations x' is made by averaging predictions from all the individual regression trees

$$f(x) = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

3.4.2.1 Random Forest

Random Forest is a model that uses the bagging technique combined with decision trees (Hyndman & Athanasopoulos, 2018), and it was first introduced by Breiman (2001). It uses a multitude of decision trees trained on random samples from subsets of the data and performs regressions on these individually. The mode of predictions from all trees is used to estimate the dependent variables on new observations (Efron & Tibshirani, 1993). If the correlation between trees is relatively low, this technique will increase performance by reducing variance.

In conventional decision trees, the number of trees is limited to prevent excessive complexity. In random forests, bagging is used to randomly select resamples of training data to split each node. This makes the model more robust with respect to noise, as it does not overfit because of the law of large numbers (Breiman, 2001). In addition to bagged decision trees, it also utilizes the Random Subspace Method (Ho, 1998), which introduces randomness by randomly sampling which predictors are used in the trees. By using random sampling, both column-wise and row-wise, it decorrelates the fitted tree models such that the variance is reduced and makes the model less prone to overfitting on training data. There are various implementations of random forest, and in our thesis, we have used the "ranger" package in R.

3.4.3 Gradient boosting

Gradient boosting is a supervised machine learning technique used in regression and classification problems. It uses an ensemble of weak prediction models, typically employing decision trees. It builds the decision trees sequentially and generalizes by allowing optimization of an arbitrary differentiable loss function. Breiman (1996) discovered that adaptively reweighing the training set, growing classifiers using new weights, and combining the classifiers constructed to date could significantly decrease generalization error.

Gradient boosting allows the use of any class of weak learners $h_m(X_i)$ to improve predictive accuracy. The weak learner $h_m(X_i)$ can take any functional form such as a GLM, a neural network, or a decision tree. Although there is no requirement for $h_M(X_i)$ to be a specific function, it is usually a tree-based learner in practice (Zhang et al., 2019). Gradient boosting combines the weak learners into strong learners in an iterative approach. In a regression problem, this is accomplished teaching model f to predict values $y = f(x)$ by minimizing the mean squared error $\frac{1}{n}\sum_i (f(x_i) - y_i)^2$. To combine several weak learners, we need to introduce a gradient algorithm with M stages. We can illustrate the principles of gradient boosting machines using the following pseudocode (Hastie, Tibshirani & Friedman, 2009):

Input to model: training set $\{(x_i, y_i)\}_{i=1}^{n}$, differentiable loss function *L(y, F(x))*, such as RMSE or MSE, with iterations *M*

1. Initialize model with a constant value:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} \mathrm{L}(y_i, \gamma)$$

2. For m = 1 to M:

1. Compute pseudo-residuals

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

2. Fitting a weak learner $h_m(x)$ to pseudo-residuals and train on training set

$$\{(x_i, r_i)\}_{i=1}^{n}$$

3. Compute multiplier γ_m by solving the following one-dimensional optimization problem:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} \mathrm{L}(y_i, F_{m-1}(x_i) + \gamma \mathrm{h}_m(x_i))$$

4. Update the model

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

3. Output prediction $F_M(x)$

In our thesis, we have used three different implementations of gradient boosting for classification and continuous outcomes, GBM, XGBoost and LightGBM.

3.5.3.1 GBM

The GBM R-package is an implementation of extensions to Friedman's gradient boosting machine and Freund & Schapire's AdaBoost algorithm. In our thesis, we use it for regression, logistic regression, and count data. The R-package was developed by Greg Ridgeway (Boehmke, Cunningham & Greenwell, 2019).

3.5.3.2 XGBoost

R's GBM algorithm supports the exact greedy leaf split algorithm. The exact greedy algorithm is computationally demanding as it enumerates all the possible splits for continuous explanatory variables. In order to do so efficiently, the algorithm must first sort the data according to explanatory variable values and visit the data in sorted order to accumulate the gradient statistics for the structure score (Chen & Guestrin, 2016).

In real-world problems, it is quite common for data input to be sparse. Sparse data is when many elements in a dataset have the value zero, or the value is missing. There are three main causes of sparsity: the presence of missing values in the data, frequent zero entries in the statistics, and artefacts of feature engineering such as one-hot encoding, which we will explain later in our method section. XGBoost implements an algorithm aware of the sparsity pattern in the data, which makes computation complexity linear to the number of non-missing entries in the input, thus reducing the resources required to run the algorithm compared to GBM (Chen & Guestrin, 2016).

Another technique introduced to improve upon the GBM algorithm is column subsampling. It is a technique used in the random forest model, but it had not been implemented in open-sourced boosting algorithms previously. Like the more traditional row sub-sampling, it aims to prevent overfitting, but also decreases computation time (Chen & Guestrin, 2016). In addition to the aforementioned modifications, XGBoost also makes changes to the system design of the algorithm. The most computationally expensive part of tree learning is to get data into sorted order. XGBoost uses column blocks, which are subsets of rows in the dataset, to enable a parallel approach to split findings. It also makes changes to how the algorithm uses cache-storing in the CPU and employ compression and partitioning techniques to increase speed (Chen & Guestrin, 2016).

3.5.3.3 LightGBM

LightGBM is an implementation of gradient boosting similar to XGBoost, but it introduces a few novel techniques to address efficiency and scalability. The main difference between the two algorithms is how they grow their trees. While XGBoost needs to scan all the data instances to estimate the information gain of all possible split points, making it very time consuming, LightGBM introduces two novel techniques (Ke et al., 2017): Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). The GOSS technique excludes data instances with small gradients, as data instances with larger gradients are more important in the computation of information gain. This means GOSS can obtain quite accurate estimations while reducing data size. EFB bundles mutually exclusive explanatory variables to reduce the overall number of explanatory variables, thus reducing computational complexity.

XGBoost has later implemented a histogram binning option to use a leaf-wise growth similar to LightGBM (Mitchell et al., 2018). This has lessened the computational time gap between the two gradient boosting implementations. LightGBM is still, however, considered the faster gradient boosting model of the two.

3.5.3.4 Hyperparameter optimization process

There are multiple ways to find the optimal hyperparameters for our models. Often it is done manually by adjusting after each run and identifying what parameters yield increased performance. It can also be automated by creating grids of different hyperparameters, which can either run through all combinations or choose a randomized approach. It essentially tests different combinations specified in a pre-determined table. In our gradient boosting models, we have used Bayesian hyperparameter optimization through the “mlrMBO”-package in R.

Bayesian hyperparameter optimization is different from other methods such as grid search and a random grid search, as it applies a probability model of the objective function and uses it to test the most promising hyperparameters. It uses past evaluation results to form a probabilistic model, mapping hyperparameters to the probability of a score of the objective function. By using a surrogate probability model of the objective function, it can run the hyperparameters that perform best on the surrogate. It then applies these hyperparameters to the objective function and updates the surrogate model with the new results. It iterates through this process until the pre-determined iteration limit is reached (Koehrsen, 2018). Using Bayesian

optimization helps reduce the number of iterations and computer usage necessary to find the near-optimal parameters, which is an important consideration in a large dataset.

3.4.4 Stacked ensemble

Stacked ensembles, also known as stacking or stacked generalization, is an ensemble technique that feeds predictions from multiple models to a new meta-model before forming the final prediction. Instead of choosing weights of different models in an ensemble, it uses their predictions as input to make predictions on the test set. This is done so that the second level model, or meta-model, can learn how the base level models may consistently correctly or incorrectly predict certain instances. It is a means of estimating and correcting for the biases of the models with respect to the provided training data (Wolpert, 1992).

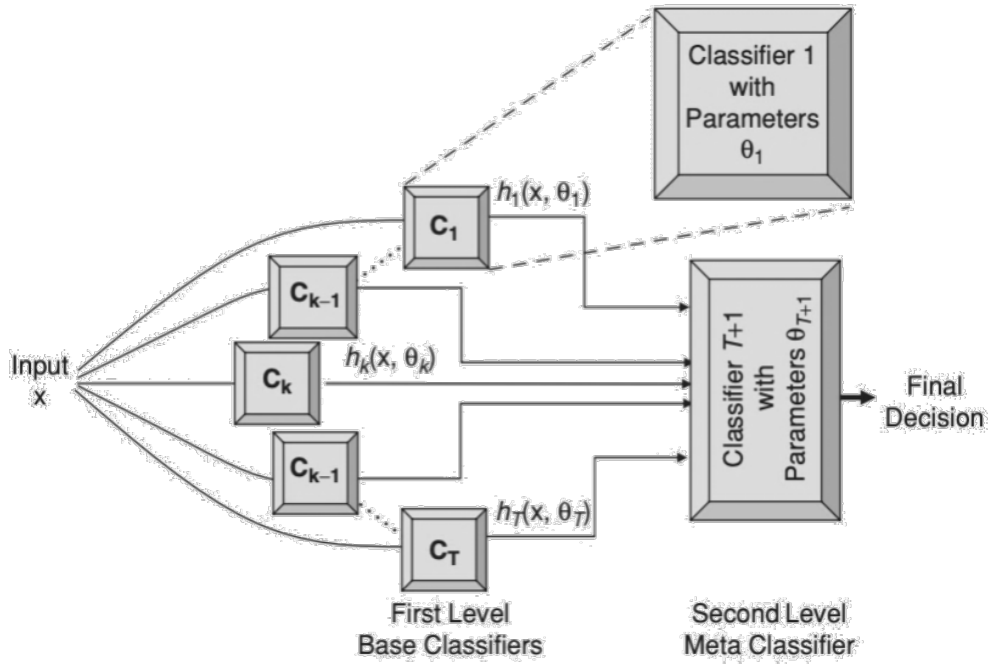

Figure 1 – Stacked ensemble overview (Polikar, 2006)

The stacked model scheme is illustrated by figure 1. Models C_1, ..., C_T are trained using training parameters θ_1 through θ_T to output predictions h_1 through h_T. The outputs of these models and response variables are then respectively used as input and output training pairs for the second level model C_{T+1}. The outputs of each model for the data subset in which it was not trained on, along with the correct labels of those instances, constitute the training data for the second level meta-model C_{T+1}. Once C_{T+1} is trained, individual models C_1, ... C_T are retrained on the training set. The same concepts of stacked ensembles are also applied to blending, which uses the same meta-model framework, but uses pre-determined training,

validation, and test sets to train the base model and meta-models instead of k-fold cross-validation.

3.4.5 Super learner implementation

SuperLearner is a framework for weighted average ensembles introduced by Van der Laan et al. (2007). It is an integrated algorithm to determine suitable candidate models in an ensemble. The candidate models are weighted by minimizing the problem loss function by using cross-validation. SuperLearner is an easy-to-use way to create ensembles in R and supports multiple well-known individual models and several methods for variable selection.

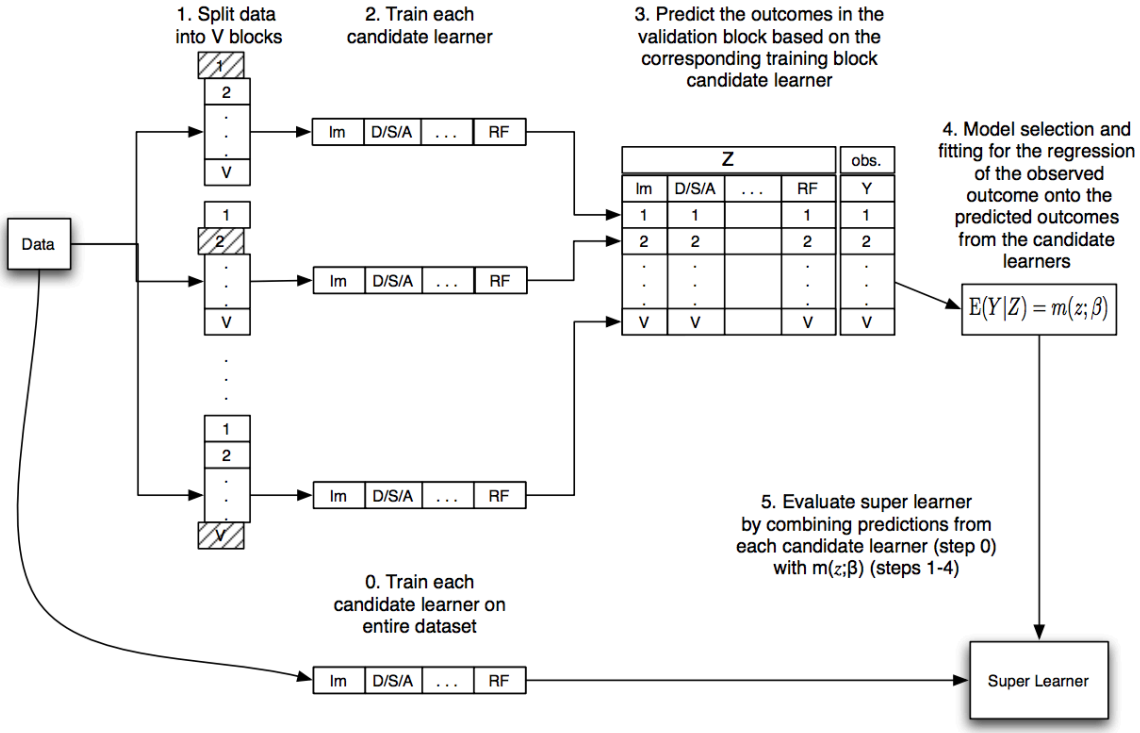

Figure 2 – SuperLearner sequence overview (Van der Laan, Polley & Hubbard, 2007)

The SuperLearner algorithm consists of six steps as illustrated in figure 2. It first splits data using v-fold cross-validation. Then it trains each pre-specified candidate model before it predicts the corresponding training block created in the first step. It then selects which models to include in the final steps based upon the performance of each individual model. The selected models are then trained on the entire dataset before using the weighted average determined by cross-validation to predict the response. In our thesis, we have primarily used the SuperLearner framework as a meta-model in a stacked ensemble.

3.4.6 Three-step ensemble

To solve the specific task of predicting the total claim of each customer, we have come up with a novel framework, that, as far as we can tell, has not been researched before. It can best be described as a combination of the principles behind neural networks and a stacked ensemble. The main target of this framework is to include information about claim propensity and claim frequency estimates to increase the performance of the total claim model. It uses model predicted responses as inputs to the next layer of training data.

We have decided to call this approach a three-step ensemble. It can be described as a way to use models to feature engineer in layers. It includes claim propensity in the first layer, before using a new training set with the added claim propensity prediction from the model trained on the first layer, to model the claim frequency. In the final layer, we train a new model including all features available in the training data, as well as the new predicted responses modelled on the first two layers for claim propensity and claim frequency, to predict the total claim. This approach makes it possible to include models specifically developed to predict claim propensity and frequency to aid the model predictions of the customer total claim size. In insurance claim modelling it is common to model claim frequency and severity separately and then combine them for the total claim estimate. In our model, we have modified this approach by including predictions of claim propensity and claim frequency as direct inputs to the total claim model.

3.5 Evaluating model performance

There are several metrics that can be used to evaluate our model performance. Earlier, we have touched upon mean squared error which is often used as a loss function for machine learning models. There is no single metric that is optimal for all situations. Different metrics have their individual strengths and weaknesses, and research indicates that using a combination of different performance metrics gives the most reliable results (Chai, 2014). We have chosen three main metrics to evaluate the performance of our models, which we will elaborate in more detail.

3.5.1 Confusion Matrix

A confusion matrix is a visual representation of the prediction performance of a classifier through a two times two contingency table. A visual representation of a confusion matrix is

illustrated in figure 3. It summarizes the correct and false predictions of the predicted class $\{N,Y\}$ on the observed class $\{n,p\}$.

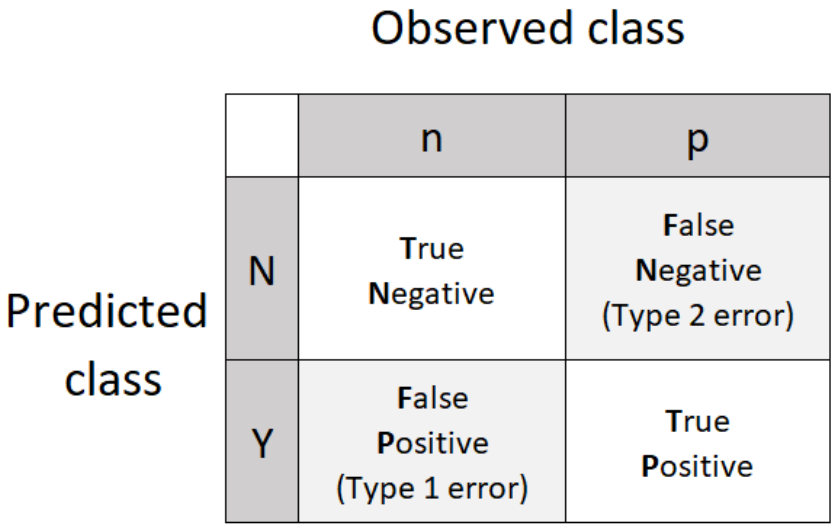

Figure 3 - Confusion Matrix

The objective of a classifier is to maximize true negative (TN) and true positive (TP), while minimizing false negative (FN) and false positive (FP). True negative is the correct classification of negative observations, while true positive is the correct classification of positive observations. False positive is the misclassification of negative observations, also known as a type 1 error. False negative is misclassification of positive observations, also known as a type 2 error. The confusion matrix is an intuitive and easily interpretable way of visualizing the ability of a model to separate between classes. A common metric derived from the confusion metric is accuracy. Classification accuracy can be calculated using the following formula.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Classification accuracy can be defined as the ratio between the number of correctly classified cases and the total number of cases (Chicco, 2020). There are, however, a few weaknesses associated with using accuracy as a metric to evaluate classification performance by itself. Accuracy does not consider that the cost of misclassification can be uneven between classes, and in situations where the dependent variable has one outcome distinctly outnumbering the other class, models are often biased towards picking the majority class (Garcia, 2010). To create a confusion matrix, the predicted class probabilities need to be converted to class responses. The appropriate threshold depends on the purpose and cost associated with

misclassifications, and the probability distributions are often different between models. Direct comparisons between models using confusion matrix-based metrics are therefore not ideal.

3.5.2 ROC and AUC

Receiver operating characteristics (ROC) graphs are a way of visualizing the ability of a model to discriminate between binary classes by varying the probability threshold between 0 and 1. The ROC graph depicts the performance of a classifier by plotting the true positive rate (TPR) against the false positive rate (FPR). The TPR, also known and sensitivity, is the proportion of people that are correctly classified as positive (TP). This is the proportion of insurance customers that have been predicted to make a claim, divided by the observed claims.

$$Sensitivity = \frac{TP}{TP + FN}$$

The false positive is equal to *1 – specificity* and is the proportion of people that were misclassified as positive (FP), divided by all observed negative cases. The false positive rate is the same as a type 1 error.

$$Specificity = \frac{TN}{TN + FP}$$

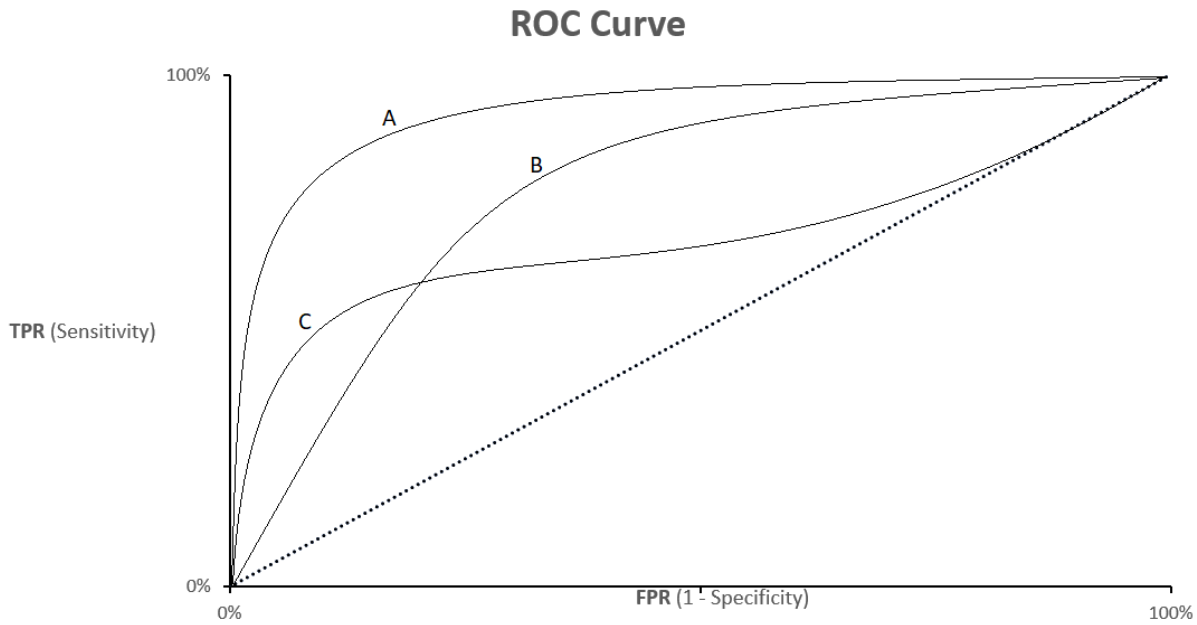

Figure 4 - ROC Curves where A dominates B and C.

The area under the ROC curve indicates how well the probabilities from the positive classes are separated from the negative classes. This area is referred to as area under the curve (ROC AUC) or area under the receiver operating characteristic curve (AUROC). An advantage of using the ROC curve is that it includes all possible classification thresholds and visualizes the prediction performance in a way that is easy to interpret. This makes the ROC AUC ideal to compare different models. The ability to compare models across different thresholds means that the ROC curve can be considered a relative operating characteristic curve because it compares TPR and FPR as the threshold is changed (Swets, 1996). When we change the classification threshold, the classifications also change. Figure 4 illustrates how ROC curve A dominates ROC curve B and C, because the AUC of A is larger than B and C for all possible thresholds (Schumann, 2002), which implies that classifier A is better than the classifier B and C, as ROC curve A has a higher AUC across all possible threshold-values. The area under A represents its AUC score, just like the area under the B and C curve represents their respective AUC scores.

AUC provides us with a score between 0 and 1, where a score close to 1 indicates a model that can perfectly discriminate between classes. The higher the AUC score is, the better the classifier is (James et al., 2017). In addition to being a great measure of performance between different models, AUC has other characteristics that make it great for our purpose. Using AUC as a performance metric avoids the need to specify the cost of misclassification (Hand, 2001), and can also be a useful tool to deal with challenges related to unbalanced data (Fawcett, 2005).

There are a few caveats associated with using AUC, which is important to keep in mind when measuring model performance. It ignores predicted probability values and the goodness-of-fit of the model, and summarises test performances over regions of the ROC space which, sometimes might not be relevant to our given problem. It also weighs omission, the fraction of values that belong to a class but were predicted to be in a different class, and commission errors, the fraction of values that were predicted to be in a class but do not belong to that class, equally. The AUC does not give information about the spatial distribution of model errors, which might be of importance in the application of the models. It is also important to keep in mind that in real-life problems binary predictions are often more important than probabilities (Lobo, Jiménez-Valverde & Real, 2008).

3.5.3 RMSE

Root Mean Squared Error (RMSE) can be used to evaluate continuous variables and represents the standard deviation of residuals. Note that the squared value of RMSE corresponds to a sample version of the mean squared error (MSE) described in section 3.1.2. and thus, measures both the bias and variance of the model. RMSE is the average squared value of the residuals of the predicted response in comparison to the observed response, and can be formulated as

$$RMSE = \sqrt{\left(\sum_{i=1}^{n}(\hat{y}_i - y_i)^2\right) / n}$$

where y_i denotes the observed values and $\hat{y}_i$ denotes the predicted values, respectively. RMSE will, by definition, punish large deviations harder than small deviations. This can make interpretability difficult, but it can also be useful if it is especially important to discourage large deviations in predicted response and observed response.

3.5.4 MAE

An alternative approach to measuring the predictive power of a continuous response is the mean absolute error (MAE). MAE represents the average absolute size of the residuals. It can be formulated as

$$MAE = \left(\sum_{i=1}^{n}|\hat{y}_i - y_i|\right) / n$$

where y_i denotes the observed values and $\hat{y}_i$ denotes the predicted values, respectively. The main difference from RMSE is that the residuals are not squared. This means that MAE does not differentiate between major and minor deviations to the same extent as the RMSE. This increases interpretability and makes it particularly useful coupled to the use of RMSE when evaluating models. An example of how using both error measures might increase insight, is when models have relatively similar MAE but distinctly different RMSE values or the other way around (Willmott, 2005). A common weakness shared by both MAE and RMSE is that they do not say anything about which direction the prediction error occurs. This makes it important to make sure models are not consistently over- or underestimating predictions.

One distinct advantage of RMSEs over MAEs is that RMSEs avoid the use of absolute value, which is undesirable in many mathematical calculations. For example, it might be difficult to calculate the gradient or sensitivity of the MAEs with respect to certain model parameters. Another issue is that many models use the sum of squared errors as the cost function to be minimized by adjusting model parameters, which means we cannot directly optimize the model for MAE. RMSE is therefore preferred over MAE when calculating model error sensitivities (Chai & Draxler, 2014).

4 Data

Our dataset contains 809 350 observations and 68 variables per observation. We can distinguish the variables provided as explanatory variables and response variables. The explanatory variables are data available to Frende at the time of determining the policy premiums of their customers, while the response variables are data available after the customer year has ended. There are 63 explanatory variables and five response variables.

	Name	Explanation		Name	Explanation
1	**Year**	Year of customer policy (2010 - 2018)	30	**Discount percentage**	
2	**Age**	Age of customer		**Vehicle safety measures**	Different classes of safety measures decided by parking (garage), GPS installation and similar
3	**Gender**	The gender of the customer	31	Maximum	
4	**County**	Home county of customer	32	Minimum	
5	**Noted customer**	Customer temporarily noted by Frende. Often due to debt collection notices, delayed payments, financial history.		**Number of policies**	
6	**Months as customer**	Months as customer in Frende	33	Life insurance	
7	**Previous departures as customer**	How many times the customer have cancelled all policies within Frende	34	Car	
8	**Customer channel**	Describes whether customer comes from a franchise, a direct customer channel or through a business partner	35	Motorhome	
8	Franchise		36	Vintage car	
9	Direct	Tends to have higher claim size and frequency	37	Motorcycle	
10	Business partner		38	Moped	
11	**Self-serviced**	Fully self-serviced customer relationship	39	Snowmobile	
12	**Yearly insurance premium**	Total premium yearly premium of all insurance policies held by customer in Frende	40	Leisure tractor	
	Insurance group	Insurance group. Expensive cars generally have higher insurance group numbers	41	Detachable caravan	
13	Maximum		42	Trailer	
14	Minimum		43	Unregistered vehicle	
	Third-party liability insurance	The minimum coverage car insurance	44	Cabin	
15	Maximum		45	Value object	
16	Minimum		46	Animal/pet	
	Partial comprehensive insurance	Medium coverage car insurance	47	Fire	
17	Maximum		48	House	
18	Minimum		49	Household goods	
	Comprehensive insurance	Maximum coverage car insurance	50	Boat	
19	Maximum		51	Family travel insurance	
20	Minimum		52	Travel insurance	
	Vehicle mileage			**Housing standard**	
21	Maximum		53	Extra high housing standard	
22	Minimum		54	Good housing standard	
	Vehicle age		55	Low housing standard	
23	Maximum		56	**Property value fully covered**	
24	Minimum		57	**Property insurance amount**	Value of insured house
	No-claims bonus	Bonus acquired by avoiding claims	58	**Flat roof**	Describes roof on house
25	Maximum		59	**Number of dorms registered on policy**	
26	Minimum		60	**Expanded house insurance**	Extra house insurance coverage
	Vehicle insurance deductible		61	**Claims registered at Frende**	Claims as customer of Frende
27	Maximum		62	**Claims registered last three years**	Prior to be being customer
28	Minimum		63	**Claims registered last three years at Frende**	Claims as customer of Frende
29	**Driver above 24**	**All drivers above 24**			

Table 1 – Overview and explanation of explanatory variables

A full overview of the explanatory variables found in our dataset is shown in table 1. We will illustrate further properties of the explanatory variables in section 4.1. The response variables found in our dataset are shown in table 2. We will illustrate further properties of the explanatory variables in section 4.2.

	Name	Explanation
1	**Claim frequency**	Number of claims made by customer
2	**Claim/no claim**	Did customer make any claims
3	**Number of products accrued**	How many policies did customer have
4	**Total claim amount**	Total amount paid to customer claims
5	**Claim percentage**	Percentage of claims to policy premiums

Table 2 – Overview and explanation of response variables

4.1 Explanatory variables

4.1.1 Customer relationship year

Frende has experienced rapid customer base growth throughout the period of our observations, as illustrated by figure 5. This means our dataset have fewer observations in 2010 and gradually more each year leading up to 2018.

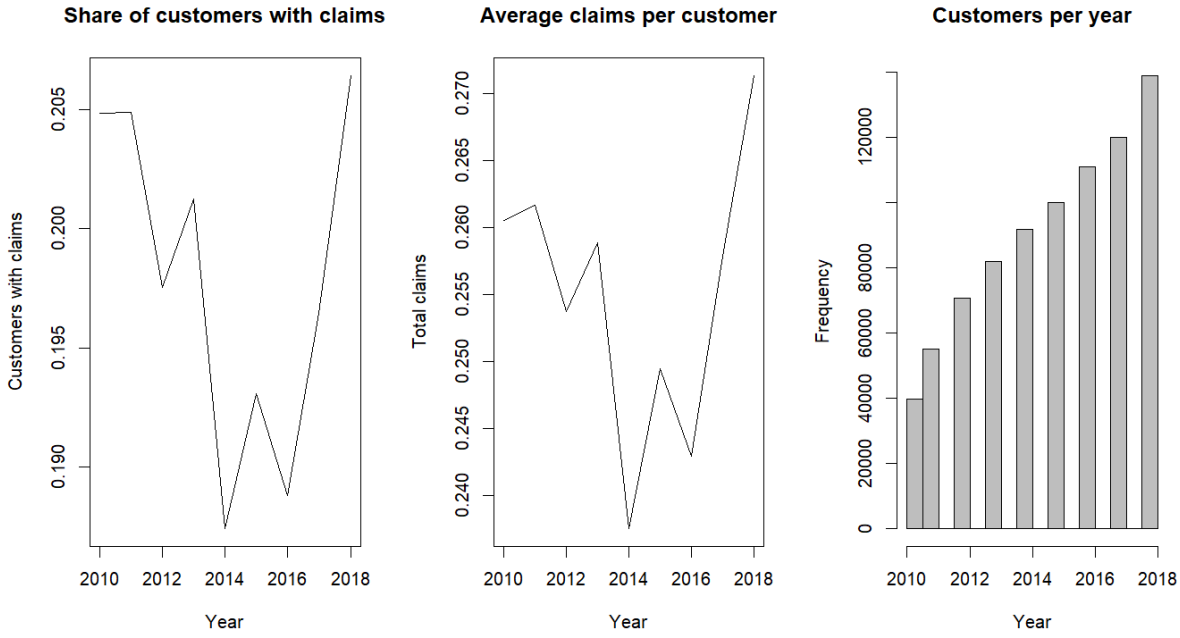

Figure 5 – Development of the yearly proportion of customers with claims, yearly average number of claims, and the number customers in Frende (2010 – 2018).

We can observe that there are several reasons why the policy year should be included as an explanatory variable. Certain years seem to have increased rates of claims, which could be in part caused by extraordinary events, such as weather conditions, pandemic outbreaks, or natural disasters. We can also expect claims to increase annually due to inflation.

4.1.2 Age distribution

Age distribution affects risk differently dependent on which insurance policies they have, and we might expect to see an increased risk from lower age groups if they have a car insurance policy but lower risk in a life insurance policy. We can observe some spikes by claim severity in different age groups in figure 6, which could be caused by a limited population of customers in that age group, or that the low and high age groups generally can be considered riskier.

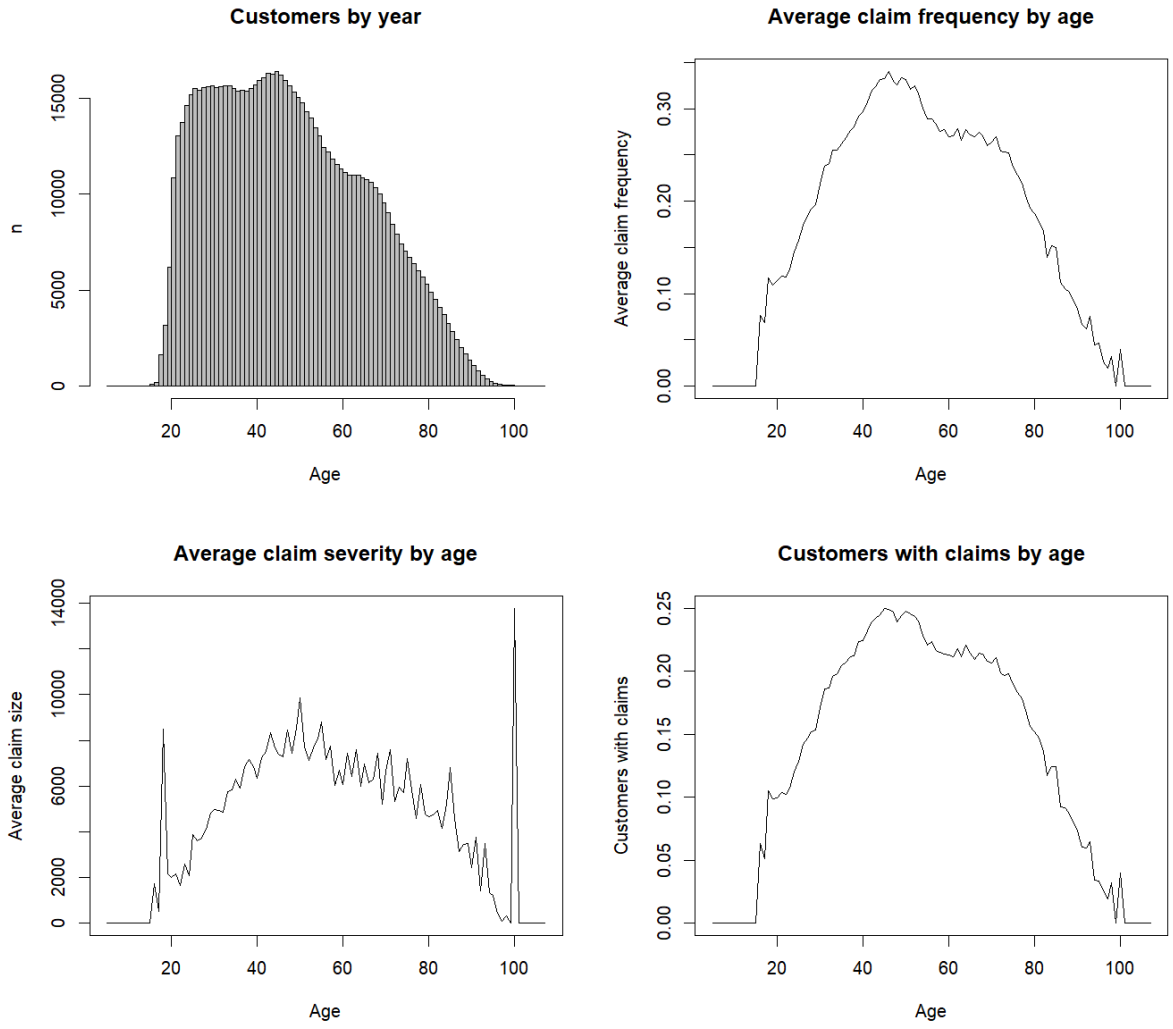

Figure 6 – Visualization of claims by age

4.1.3 Gender

While the European Economic Area (EEA) rules state that discrimination between men and women in access to goods and services is forbidden, Norway among other countries applied an exception to this rule for the insurance industry. In 2011 an EU court found that this practice is not valid, and the Norwegian government adopted a new legislative framework in 2014 mandating gender-neutral private policy premiums in Prop. 87L (Det kongelige finansdepartementet, 2014).

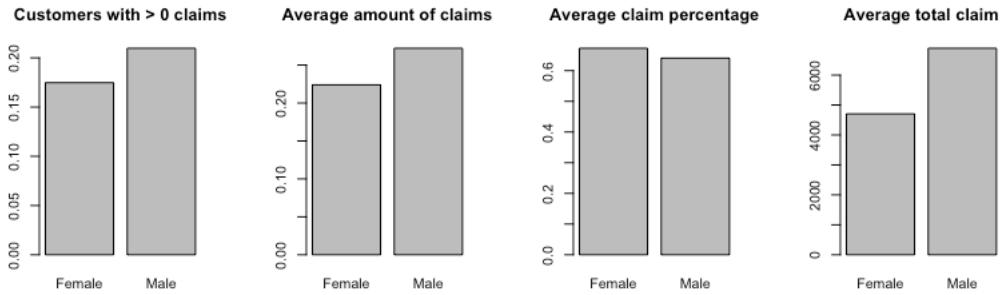

Figure 7 – Visualization of claims by gender

Differences between the genders are shown in figure 7. Males are overrepresented among customers making claims, generally have higher claim frequencies, and have more severe claims. We do, however, see that when adjusted to their overall policy premiums, men have a slightly lower average claim percentage.

4.1.4 Counties

Frende has significantly more customers along the west coast of Norway. In the duration of our observations, they have seen growth in other geographic areas, but still have their majority of customers centered around the west coast. Different areas tend to have different exposure to extreme weather, landslides, and floods. Socioeconomic conditions might also vary across customers according to the county registered in their insurance policy. The claim propensity distribution of all counties is shown in figure 8.

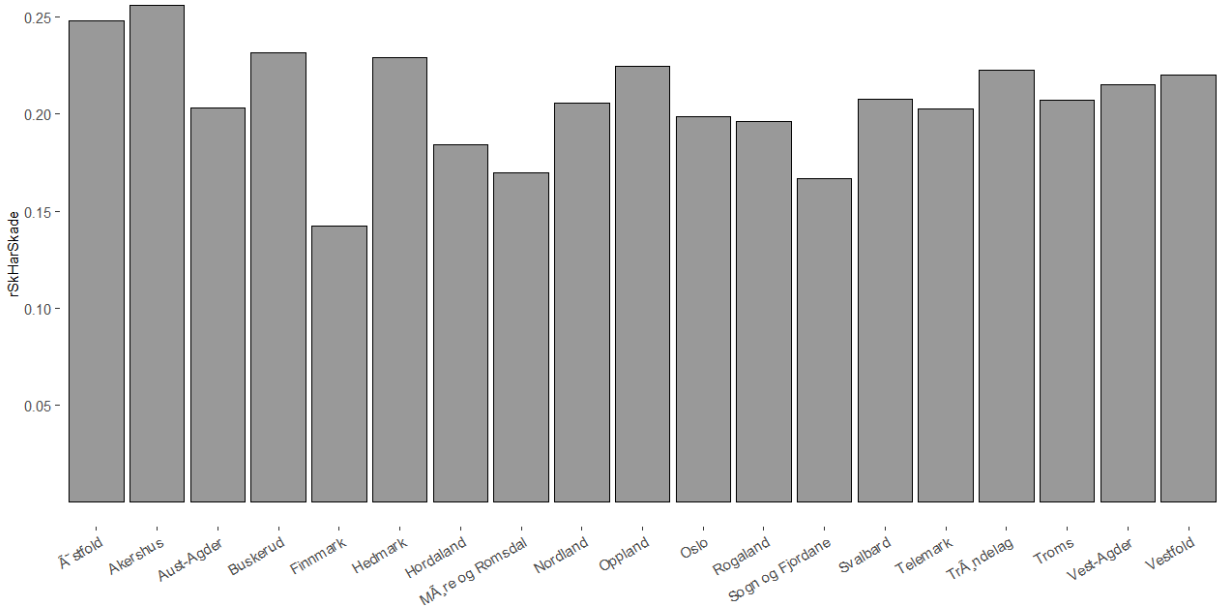

Figure 8 – Customer claim propensity by county

4.1.5 Noted customers

Frende has a system to mark customers which for various reasons are considered riskier. This could be a previously high claim frequency, failing to pay for policies on time or that they have received collection notes. The financial and non-financial conditions that cause customers to be noted can also, when improved, lead to the removal of such customer marks. From figure 9 we can see that 4.06 % of all yearly customer relationships have been noted.

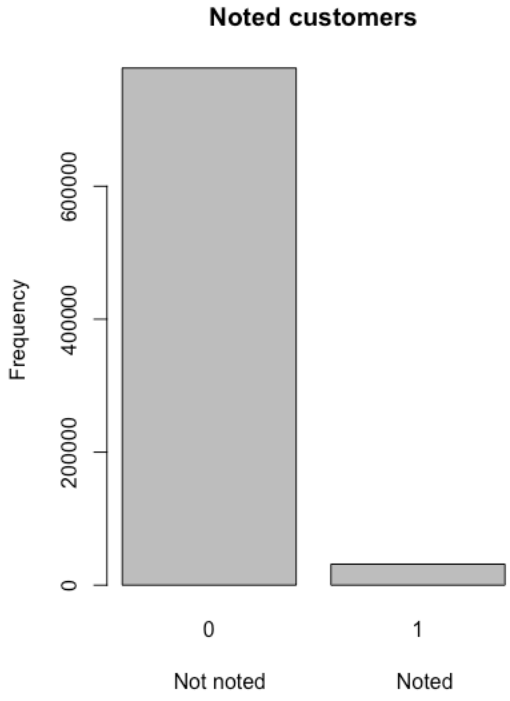

Figure 9 – Number of noted customers

The customers who are noted displayed an overall lower claim propensity than non-noted customers on average, which could be considered surprising. However, we do see that noted customers on average have a higher claim frequency than non-noted customers, with 0.262 claims per customer per year compared to 0.255 claims per customer per year for non-noted customers. This means that the noted customers are more likely to have multiple claims than non-noted customers. The average claim amount is also higher, with 7 464 NOK per noted customer compared to 6 057 NOK per non-noted customer.

4.1.6 Customer relationship length and previous departures

Figure 10 shows the number of departures of customer relationships in earlier years, i.e. the number of times a customer previously has left Frende, either by cancelling their customer relationship or by moving to another insurance provider. We can see that the vast majority of yearly customer relationships have not had previous departures, while a small minority have frequent departures.

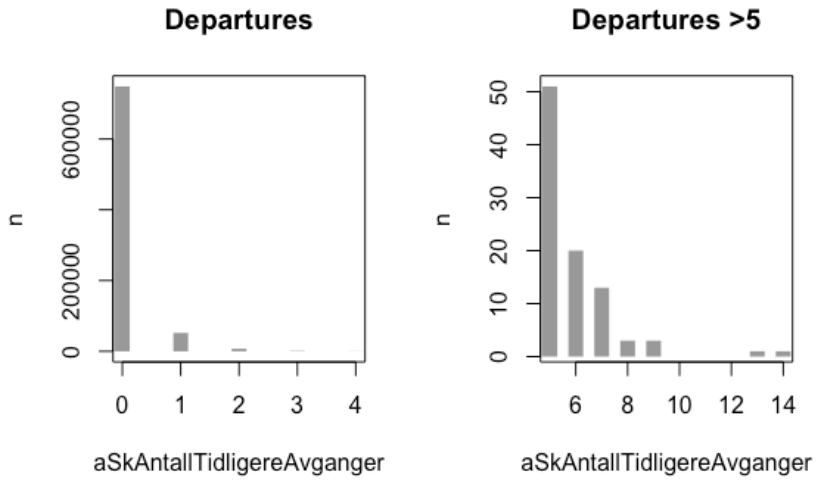

Figure 10 – Previous departures among customers

The overall customer relationship length is illustrated by figure 11. Most customers at Frende are relatively new, and only a few long-standing customer relationships have accrued since the insurance company was started.

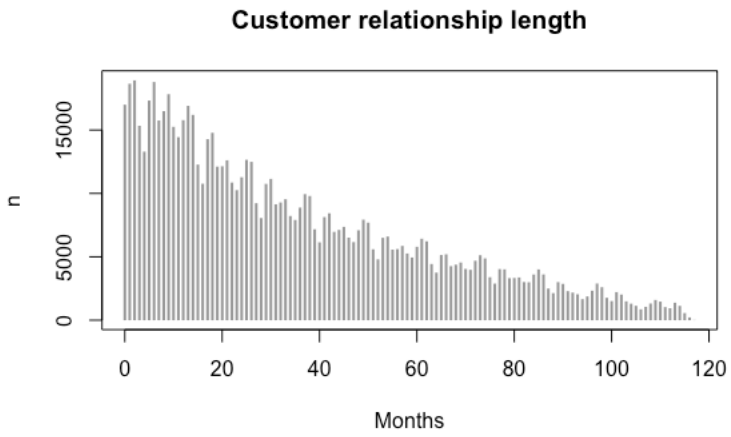

Figure 11 – Customer relationship length

4.1.7 Policy overview

While each customer can only have one life insurance, travel insurance, and family travel insurance product, customers can have multiple policies in other product categories such as car insurance and fire insurance. Figure 12 shows how many policies each customer holds in each category. It also shows that inventory, housing, car, life, travel and, family travel insurance are the most common products held by its customers.

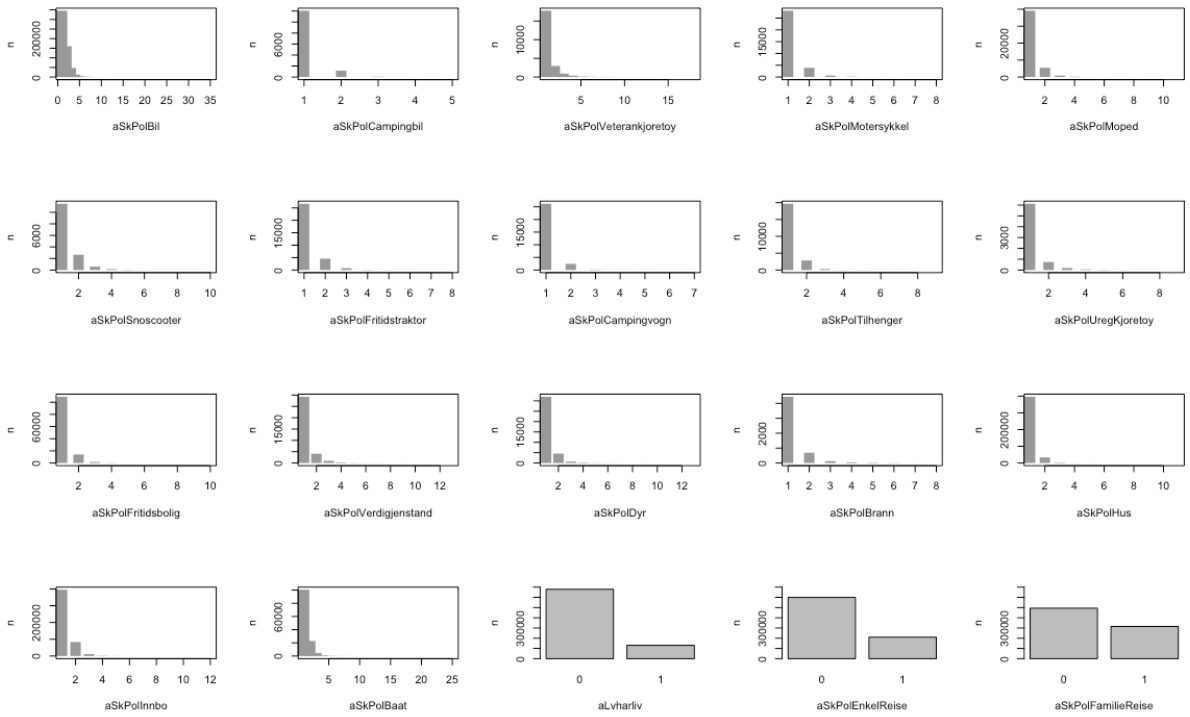

Figure 12 – Insurance policies held by customers

The policies all have different characteristics, where some have more frequent claims, and some have more severe claims. House insurance claims tend to be more severe than animal insurance claims, for example.

4.1.8 Policy premiums

The aggregated policy premiums consist of the combined yearly premiums of all policies held by customers throughout the year. It is determined by well-proven models employed on the individual policy level. Figure 13 provides insight into the distribution of yearly premiums across years and customers.

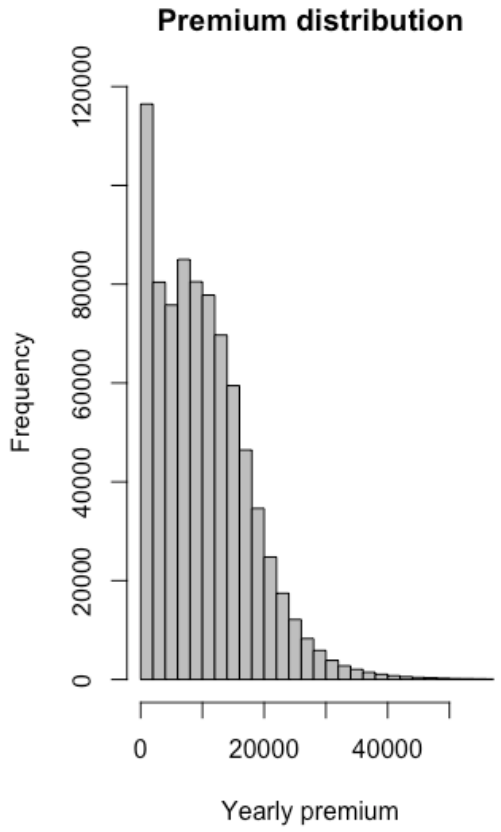

Figure 13 – Premium distribution

The mean total customer policy premium is 10 300 NOK, while the maximum total customer policy premium was 198 324 NOK. The policy premium is determined by risk factors weighted by the individual policy models of Frende, but also by the number of policies the customer held in the withstanding year. We can see that most customers pay low yearly premiums, while fewer pay large yearly premiums.

4.1.9 Customer channel

Frende operates different channels in which insurance products are sold. While many policies are sold directly from Frende, they have a franchise channel in which third-parties sell on their behalf. One might expect different customer characteristics based on what channels customers are obtained from. We can see how each customer relationship year is distributed between channels in figure 14.

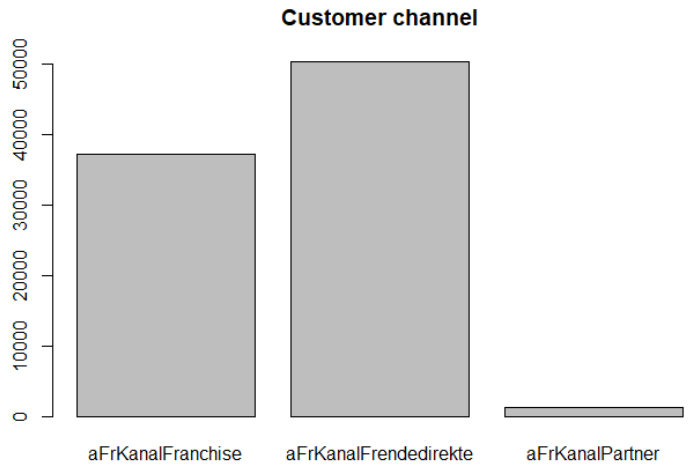

Figure 14 –Customer channel

Most observations are not registered with either customer channels, which is caused by the dataset not having an explicit variable indicating customers gained through the co-operating banks selling insurance policies on behalf of Frende. Customers obtained through these banks make up the vast majority of their customer base throughout the period from 2010-2018.

4.1.10 Former claims

In addition to current customer data, we have access to historical data showing how many claims the customer has made the last three years as a customer at Frende. From figure 15, we can see the number of customers that have had previous claims before their current policy year.

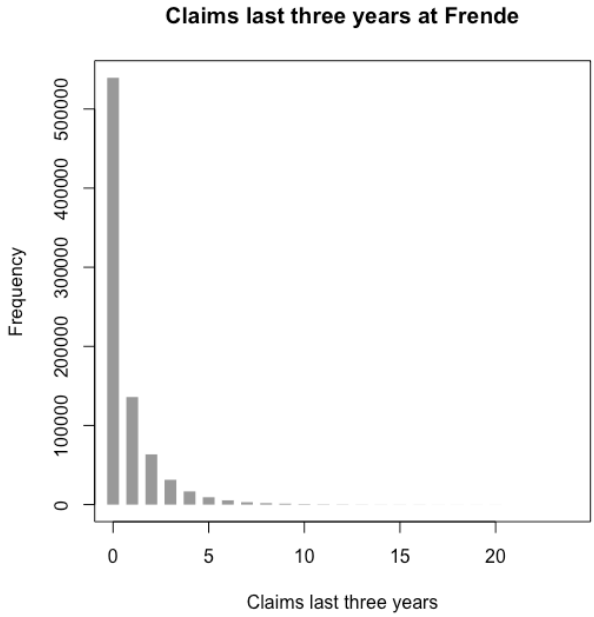

Figure 15 – Claims last three years at Frende

In addition to historical claims data on their existing customers, Frende has access to claims made by the customer for the last three years in other insurance companies. We can see the number of customers who have and have not made claims three years prior to signing an insurance policy at Frende in figure 16.

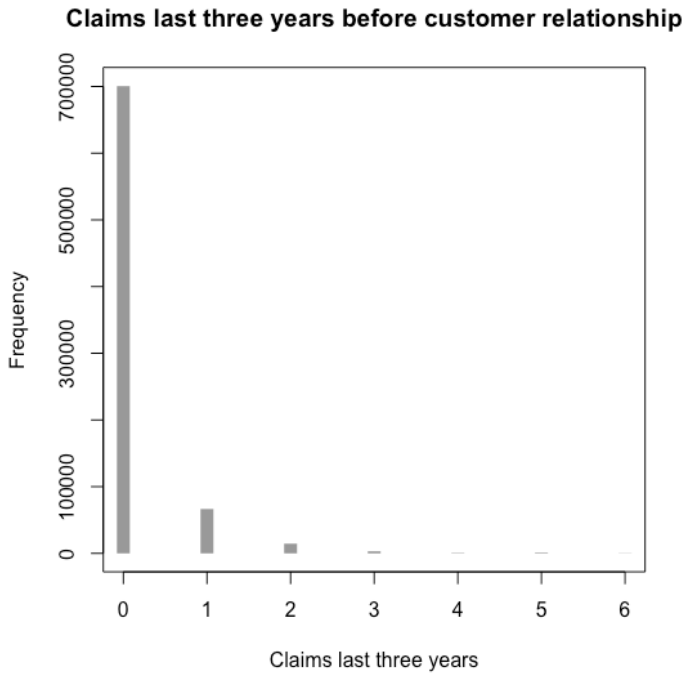

Figure 16 – Claims last three years before customer relationship

4.1.11 Housing insurance

We have access to several explanatory variables associated with housing insurance. Figure 17 shows the housing standard, which is divided into low, better, and extra high.

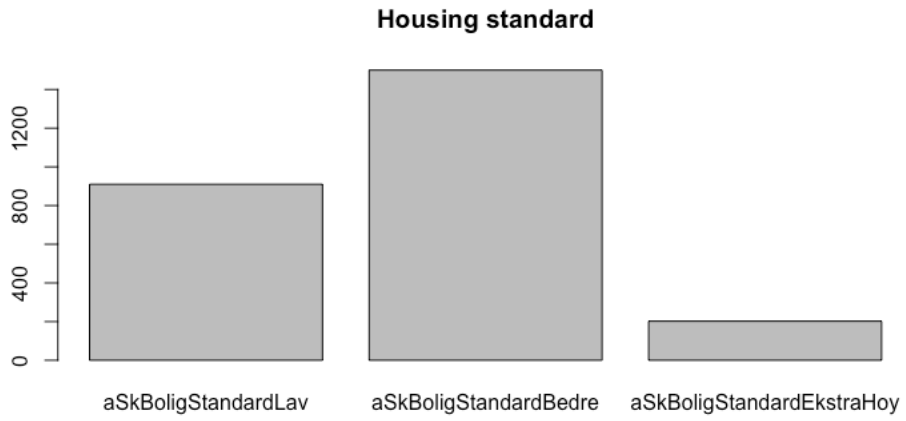

Figure 17 – Housing standard

In addition to the overall housing standard, we have explanatory variables shown in figure 18 indicating if the house has extra coverage, a flat roof, whether it is insured at its full value, and the number of dorms included in the housing policy.

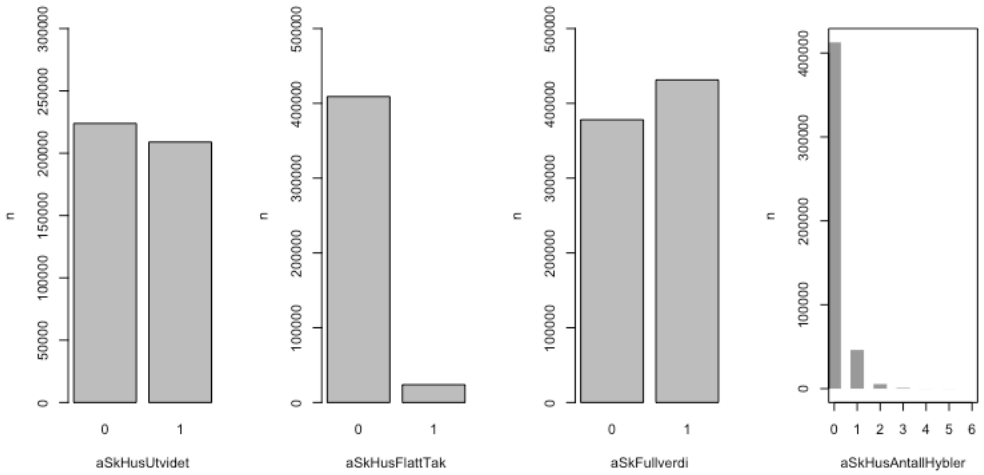

Figure 18 – House insurance policy variables

Finally, figure 19 shows the distribution of insurance values for the houses. It is close to normally distributed with a mean of 2 123 101 NOK, but with a heavier right tail. The lowest housing insurance sum is 10 000 NOK, and there are houses insured for up to 26 653 550 NOK making the insurance value a long-tailed distribution.

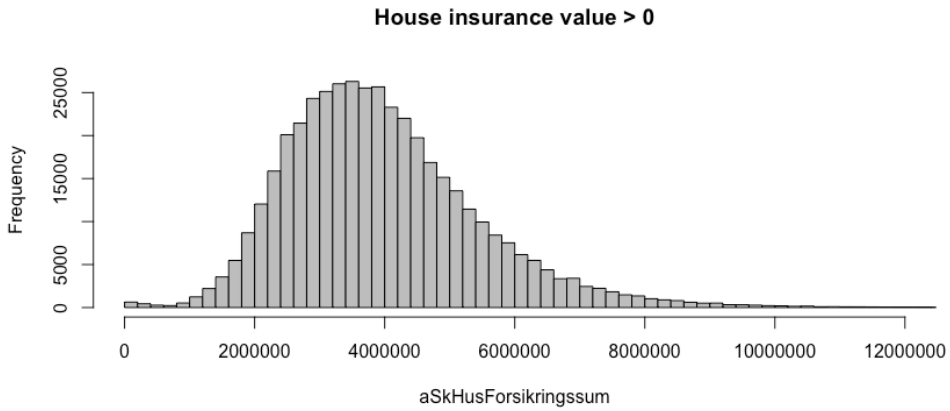

Figure 19 – House insurance value

4.1.12 Car insurance

Just like the housing insurance data, we have access to additional explanatory variables for car insurance. As each customer can have multiple car insurance policies, we have two different numbers for each variable, giving the minimum and maximum value. As most customers have only one car insurance policy, these numbers tend to be quite similar. The only exceptions to this structure are the insurance discount and minimum age variables, which only has one singular value each. Figure 20 illustrates the car insurance data available.

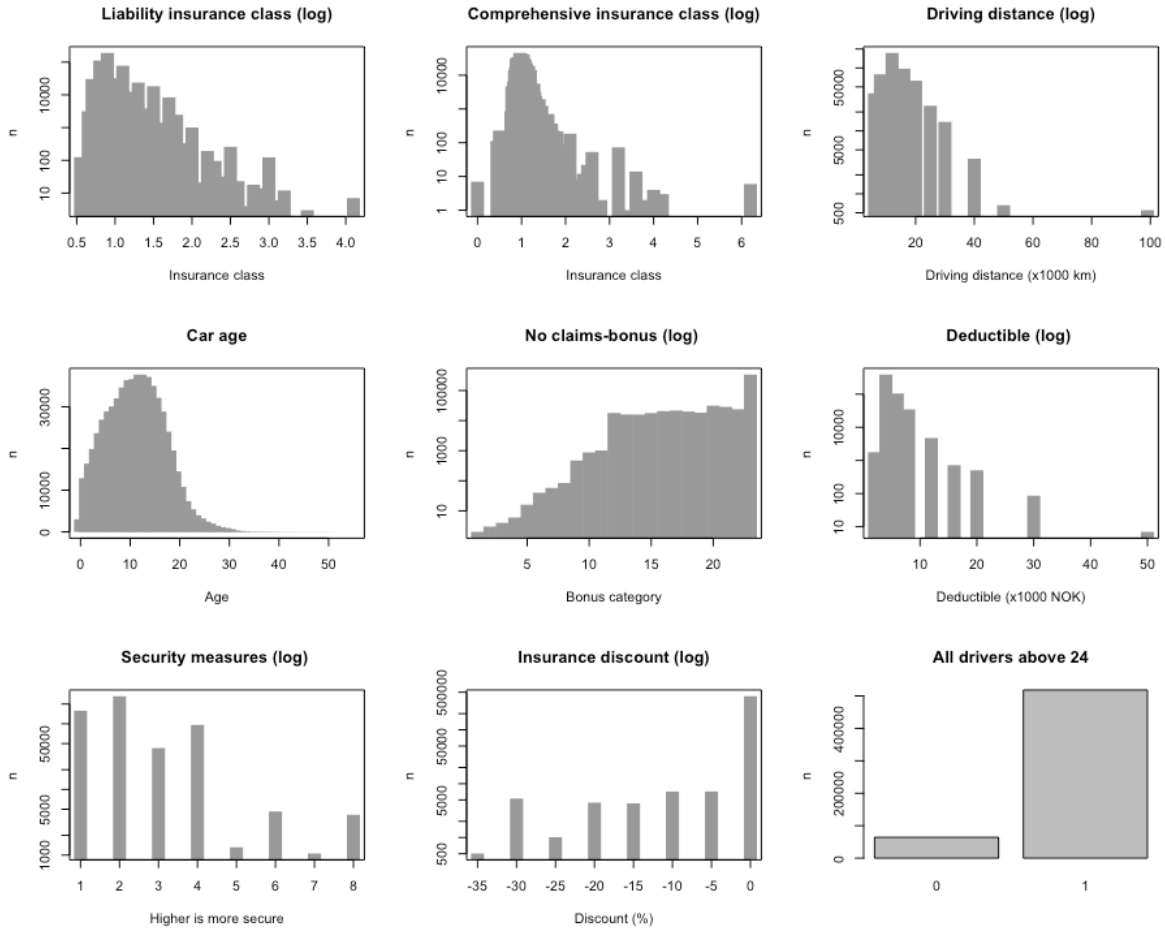

Figure 20 – Car insurance policy variables

4.1.13 Covariance between predictors

Figure 21 shows the covariance between most of our explanatory variables, with some exceptions. The factor variables of county and customer year are highly correlated and therefore omitted. The variables "aSkPolVeterankjoretoy", "aSkPolBrann", "aSkPolCampingbil", "aSkPolDyr" and "aFrMann", have very low correlation with the remaining 56 explanatory variables and were omitted to make the plot easier to read. The covariance matrix shows a high positive correlation between the car insurance predictors. This is to be expected as customers without car insurance will have no attributes available and will be set to zero, while customers with one or multiple car insurance policies will automatically have a high correlation between this group of explanatory variables. This also seems to be the case with the additional house insurance predictors.

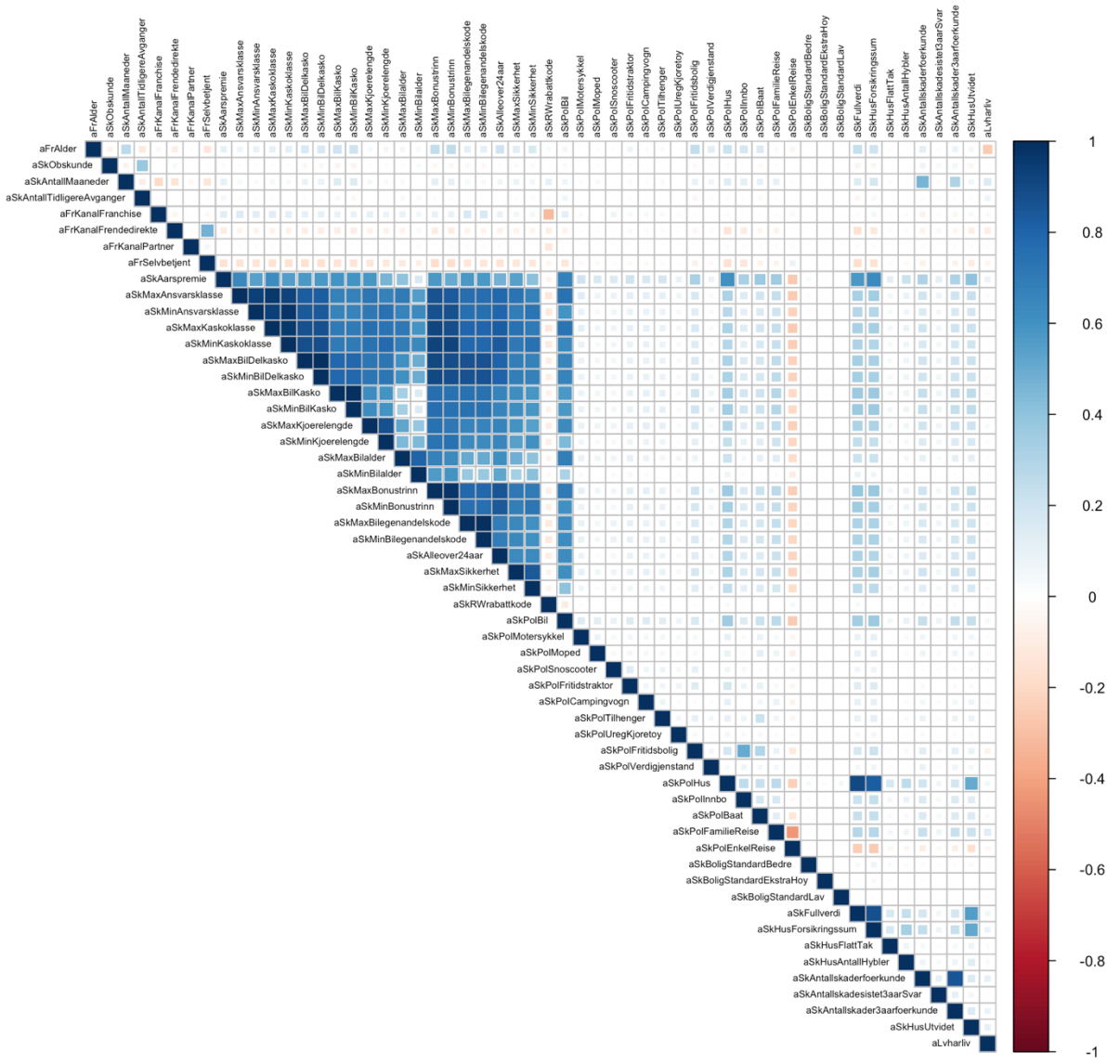

Figure 21 – Explanatory variable covariance matrix

The figure also shows that yearly insurance premium (aSkAarspremie) positively correlates with most predictors with the exception of discount code (aSkRWrabattkode) and individual travel insurance (aSkPolEnkelReise). The car insurance discount is positively correlated with customer insurance premium, which is caused by the discount being a negative number, and individual travel insurance is mildly negatively correlated with car insurance policy (aSkPolBil) and, therefore, also the other car insurance predictors. As these are among the most popular policies offered by Frende and customers tend to have either car insurance or individual travel insurance, but it is rarer to have both. This also holds true with house insurance and individual travel insurance. There are some highly negatively correlated predictors. Usually, this indicates that the insurances are mutually discriminatory. This is often caused by policies or customer attributes to be either unlikely or impossible to combine. Two examples of this are that the different customer channels are mutually exclusive, while family travel insurance policy is unnecessary to combine with an additional individual travel policy.

4.2 Response variables

Our dataset contains five response variables. In our thesis, we have modelled the binary claim variable, the claim frequency variable, and the total claim amount variable. Figure 22 shows a visual representation of these response variables.

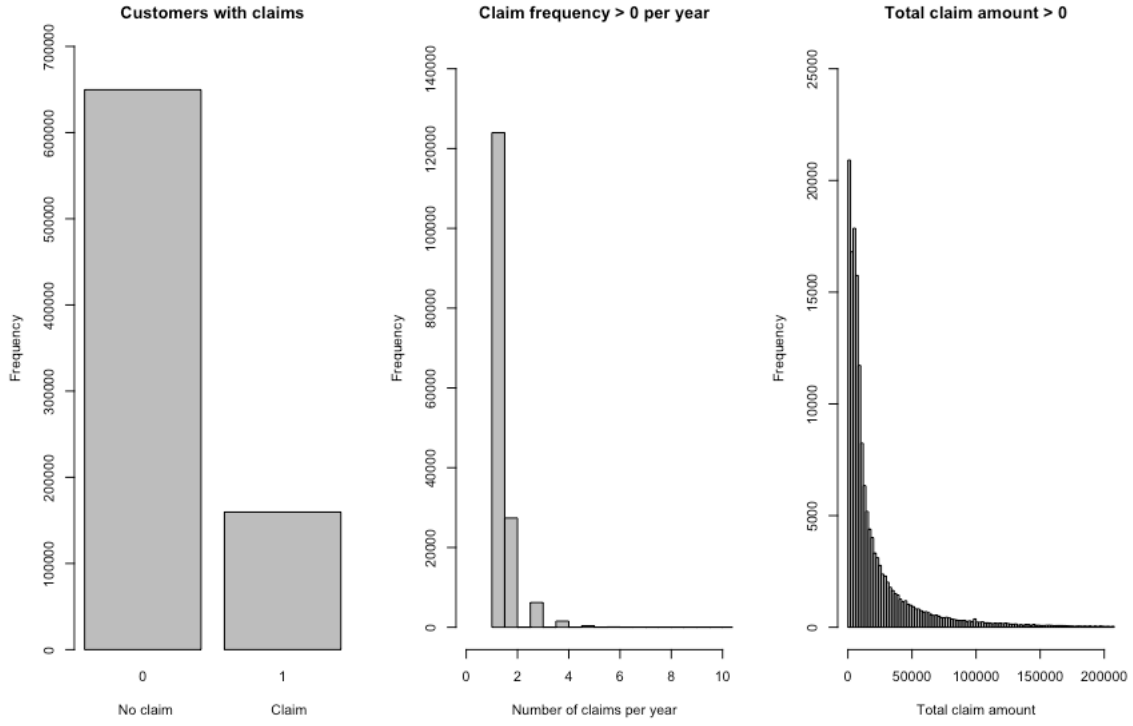

Figure 22 – Customer claim overview by claim propensity, frequency and total claim size

4.2.1 Claim or no claim

The majority of Frende customers do not make any claims on a yearly basis, as shown by figure 22. On average, 19.72 % of Frende insurance customers make at least one claim per year. This makes the response variable somewhat unbalanced. We did not, however, find it so severely unbalanced that we need to use sampling techniques to improve predictions.

4.2.2 Claim frequency

Figure 22 also shows that among customers with claims, most have relatively few claims. There are some customers with high claim frequencies, but they make up a very small part of the customer base. There is a steep reduction in yearly customer relationships that have many claims. To give a representative picture of the claim frequency, we decided to adjust the claim frequency by dividing it by products accrued. This means we have a claim frequency adjusted by the accrued products throughout the year. The new adjusted claim frequency variable is visualized in figure 23. From this point on in the thesis, claim frequency will refer to the new adjusted variable.

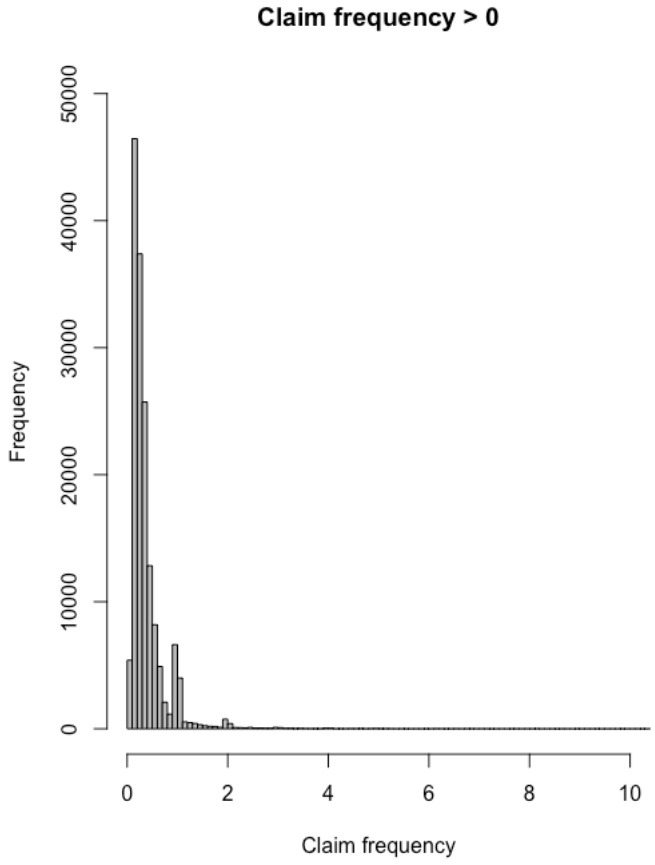

Figure 23 – Adjusted claim frequency response variable

4.2.3 Total claim amount

The total claim amount response variable is the aggregated claim amount per customer per year over all insurance policies held by that individual customer, and its distribution is shown in figure 22 and 24. The mean of all claims is 6 111 NOK, while the median is 0. If we look at the actual claims, meaning all observations above 0, we get a median of 9 400 NOK and a mean of 30 986 NOK. This indicates that most claims are smaller than the mean average and that there are some quite large claims that greatly affects the mean average. This is underlined by the smallest registered claim in the dataset being 501 NOK and the largest registered claim being 17 136 218 NOK. To see which distribution best fits the total claim variable, we can look at figure 24, showing the observed distribution along with the continuous distributions described in section 3.2 fitted to the data.

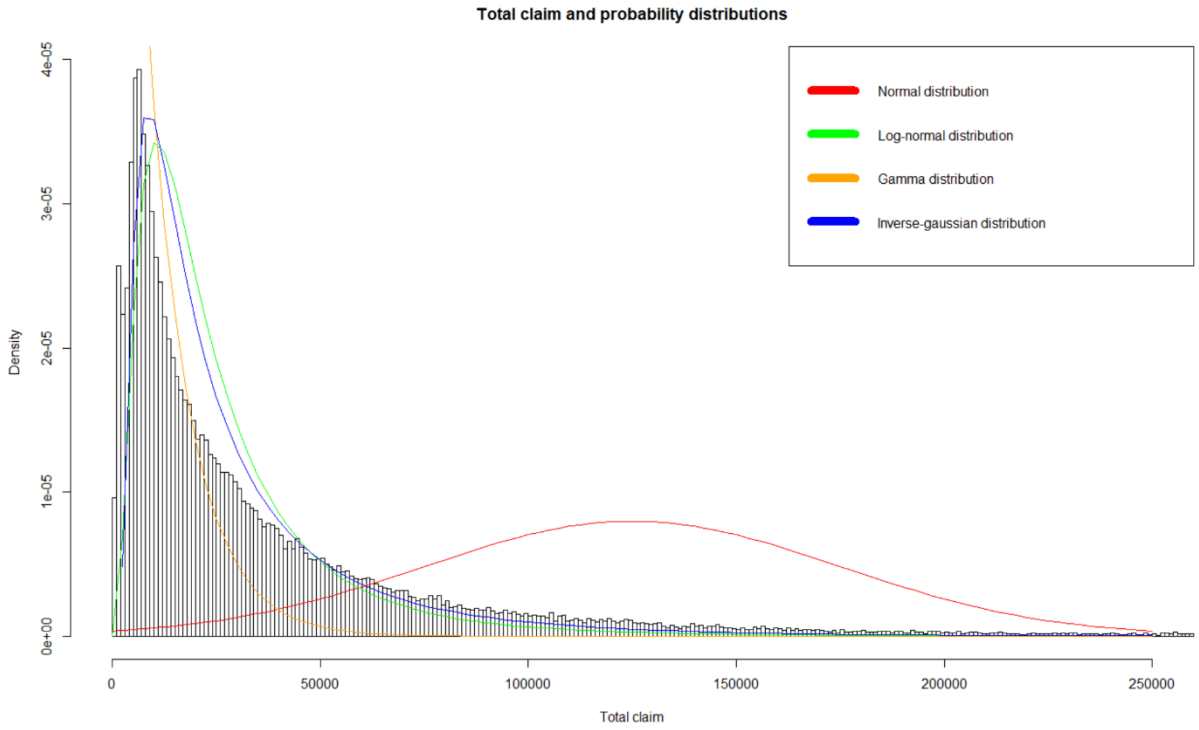

Figure 24 – Observed claim size compared to probability distributions

Figure 24 indicates that the distributions for gamma, log-normal, and inverse-gaussian all share similar properties to the observed distribution. The distributions closest to the observed distribution are the inverse-gaussian and log-normal distributions.

4.3 Feature engineering

Some of the machine learning algorithms used in our thesis only accept numerical variables, which means factors need to be represented as numeric values. It is also important to ensure that our models do not misinterpret the mathematical relationships between an explanatory variable and the response variable by asserting meaning to different factor levels when they do not (Aggiwal, 2017). Examples of such factors in our dataset are the year and county variables, in which the numeric representations do not have any mathematical relationship. To avoid such an inaccurate interpretation of the data, we have hot encoded these variables. It is a method to represent categorical variables as binary vectors. This means that the hot encoded variable is split into dummy variables for each level of the variable. The original variables are then removed to avoid multicollinearity with the new dummy variables (Mahto, 2019).

5 Modelling customer claim risk

We refer to customer claim risk as the risk of a customer making one or more claims, regardless of its severity and frequency. This can be useful to discover certain customer traits that increase the claim propensity. We will present the results achieved by our models, but also which customer traits the models deem most important to predict if they will make a claim or not.

5.1 Model predictions

We will use AUC to compare the performance of our models, using the results of all available customer data, but also using the yearly premium as the only explanatory variable. This is done to see if the additional individual customer data results in improved predictions.

5.1.1 AUC comparison

We can see significant differences between the models using all the variables, compared to those who only use the yearly premium. Even the best model using the yearly premium, the stacked ensemble, only manages an AUC of 0.6879, which is worse than the worst-performing model using all data, the random forest, with an AUC of 0.7008. This shows the usefulness of additional individual customer data. The overall results of all models can be seen in table 3.

Model	AUC (All variables)	AUC (Yearly premium)
Neural Network	0.7247	0.6856
GLM	0.7265	0.6856
LightGBM	0.7288	0.6853
XGBoost	0.7299	0.6856
Random Forest	0.7008	0.6768
Averaging	0.7274	0.6870
Stacked Ensemble	0.7301	0.6879

Table 3 – Model AUC results comparison

Table 3 shows that the best performing model is the stacked ensemble with an AUC of 0.7301. The stacked ensemble uses the other models shown in the table as its inputs. We can see it outperforms the XGBoost model, which is the best-performing singular model, with an AUC of 0.7299. It is closely followed by the similar LightGBM model with an AUC of 0.7288 and a GLM with a ridge penalty term achieving 0.7265. The neural network achieves a

score of 0.7247, outperforming only the Random Forest model at 0.7008. Figure 25 provides a detailed view of the ROC curves of the models.

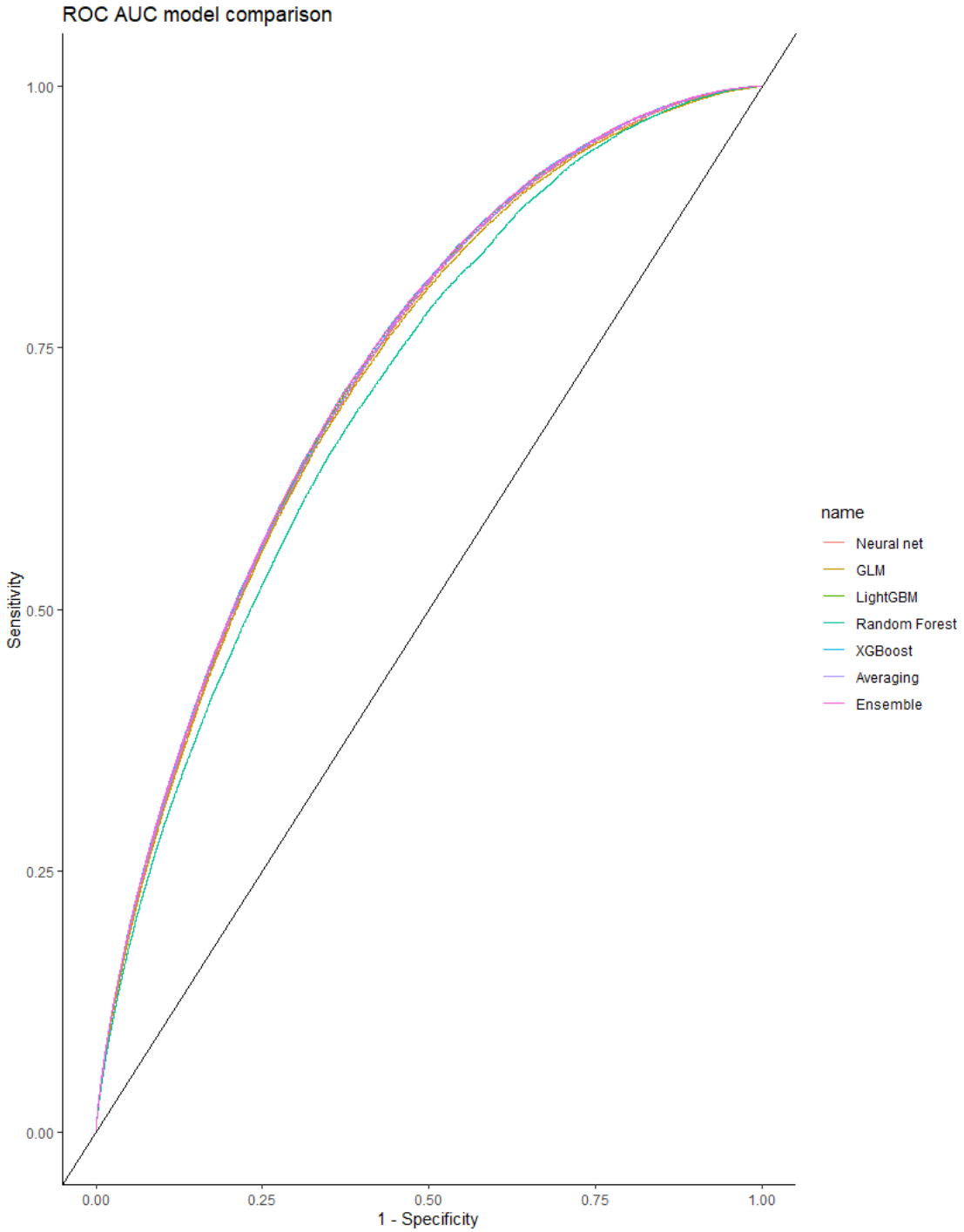

Figure 25 – ROC AUC model comparison using all individual explanatory variables

The big difference between predictions using the full set of explanatory variables and the yearly premium was rather unsurprising. The individual data contains, as described in the data section, a lot of information about the policies each customer hold. We would expect yearly premiums to not only be determined by the claim propensity of a customer, or the expected

claim frequency, but also the theoretical claim severity if a claim occurs. While travel insurances typically result in high frequency claim patterns, the typical claim is relatively small. Housing insurance is expected to have opposite properties, with claims being rare but often quite severe if they occur. By including additional data about the different policies held and customer characteristics, the model should be able to better distinguish insurance policies which significantly increases claim propensity and policies that are less likely to do so.

We can also observe that the averaging and stacked ensembles still perform particularly well when using only yearly premium as an explanatory variable. This indicates that these techniques are not only helpful for performance but also show stability by performing well when removing explanatory variables. The stability reveals how the model detection accuracy is affected by using different variables in testing and training data, and the variation of the size of training data (Lin et al., 2017).

5.2 Effects of explanatory variables on customer claim risk

The random forest and gradient boosting models XGBoost and LightGBM can provide variable importance. The variable importance is a measure of how important each explanatory variable is to determine the outcome of the response variable. The stacked ensemble uses the other model's predicted response variables as explanatory variables, and will therefore not have any meaningful relation to the variable importance of each underlying model. The neural network is theoretically possible to visualize, but the complexity and built-in feature engineering mean the variable importance output is often not very useful to compare how each variable affects model performance. We have decided to use the variable importance from the random forest and gradient boosting models for our variable importance plots.

Figure 26 shows that the random forest model assigns the greatest importance to family travel insurance, various car insurance variables, housing insurance variables, and customer history. The random forest model uses the Gini impurity to determine variable importance. While it is important to note that Random Forest was our worst performing model, it is often useful for visualization purposes as its underlying decision tree algorithm means it is among the more interpretable models used in our thesis.

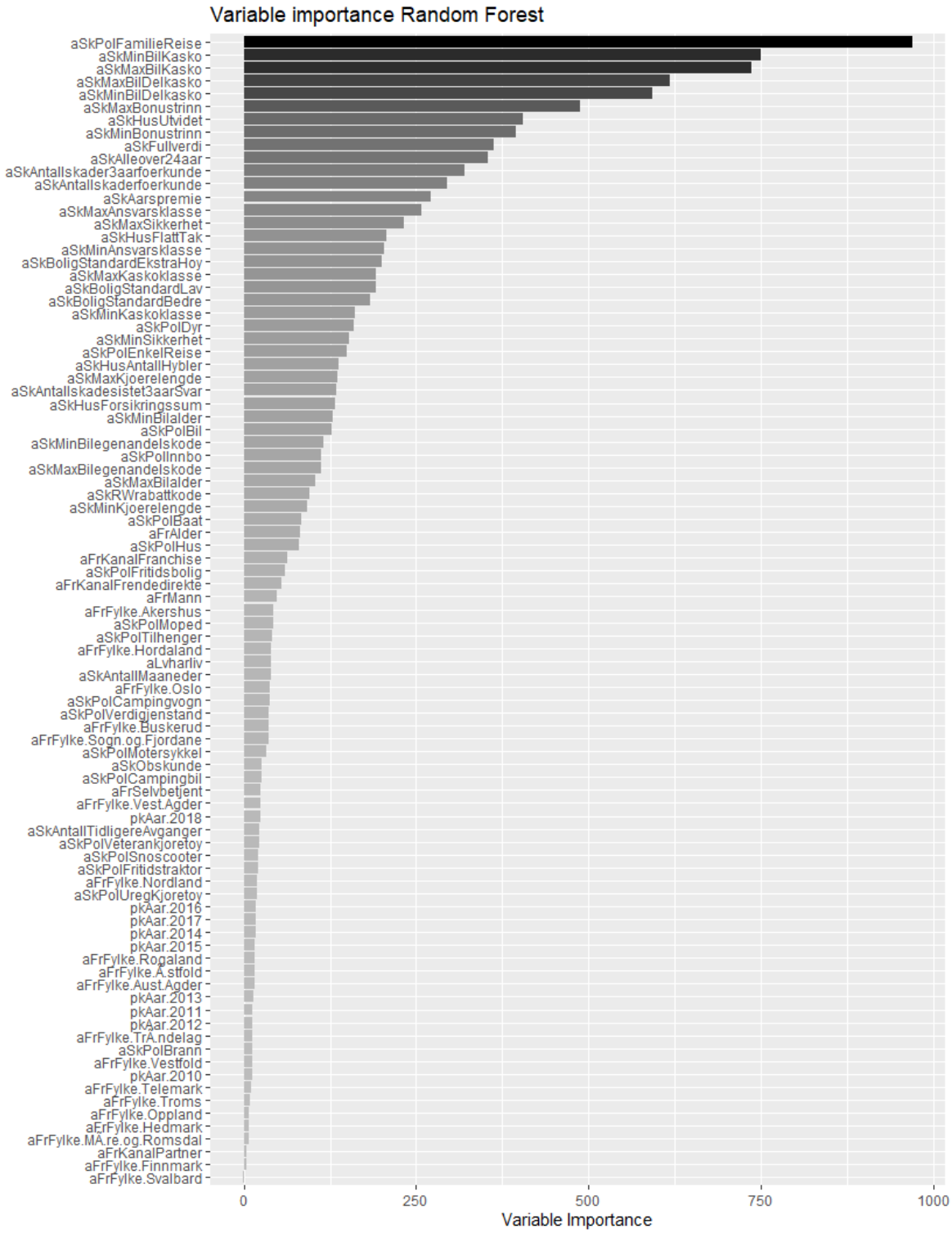

Figure 26 – Claim propensity variable importance using random forest

Figure 27 shows the variable importance of our best-performing model, XGBoost. It measures the gains made in each leaf split by variable. While the XGBoost shares previous customer history, car insurance variables, and family travel insurance as some of its more important variables, it puts greater emphasis on yearly customer insurance premium relative to the random forest model.

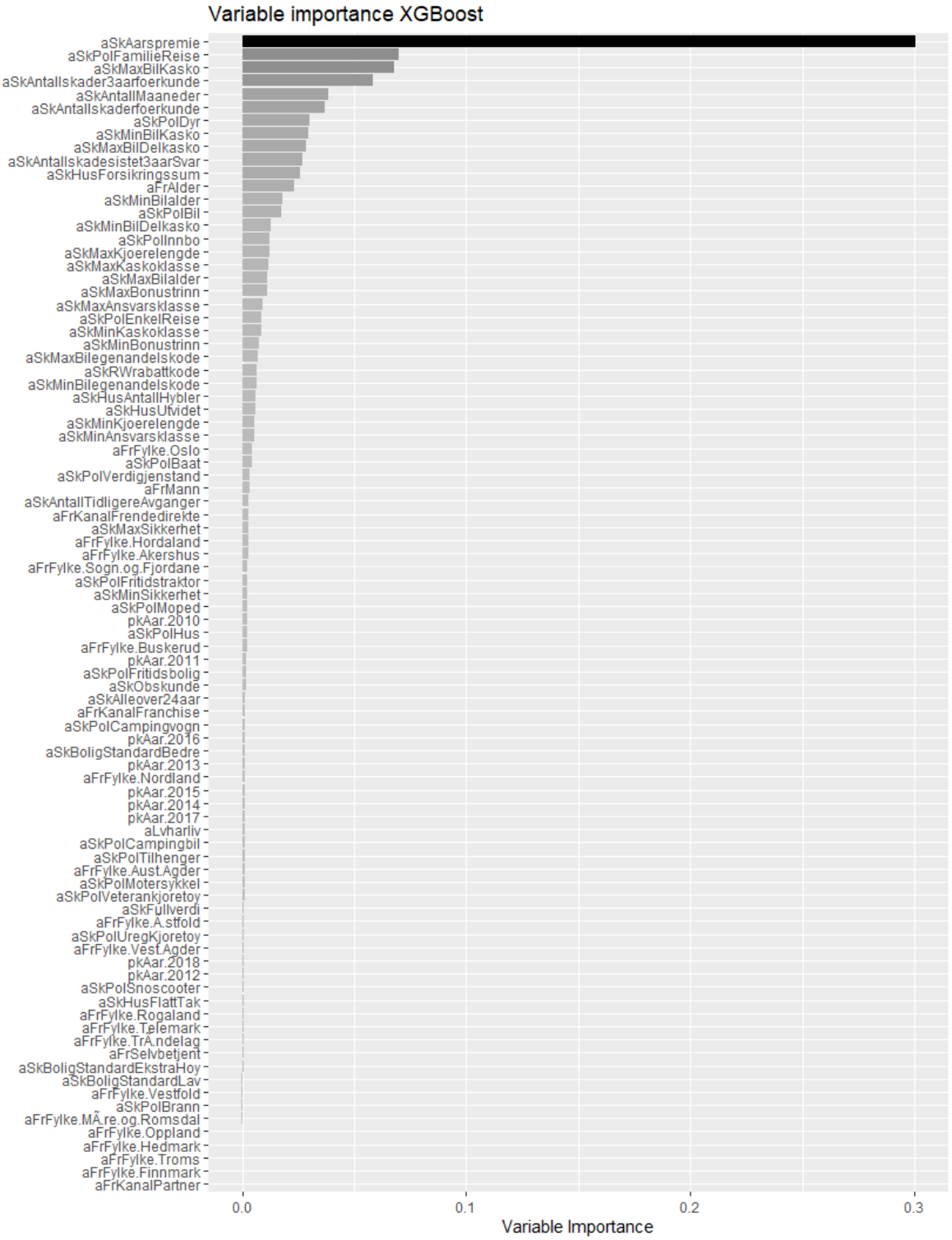

Figure 27 – Claim propensity variable importance using XGBoost

We would expect the largely similar LightGBM model to assign similar variable importance compared to the XGBoost model. Figure 28 illustrates the similarity but also shows some minor differences in the ranking order. LightGBM for example puts more relative emphasis on customer relationship length than XGBoost.

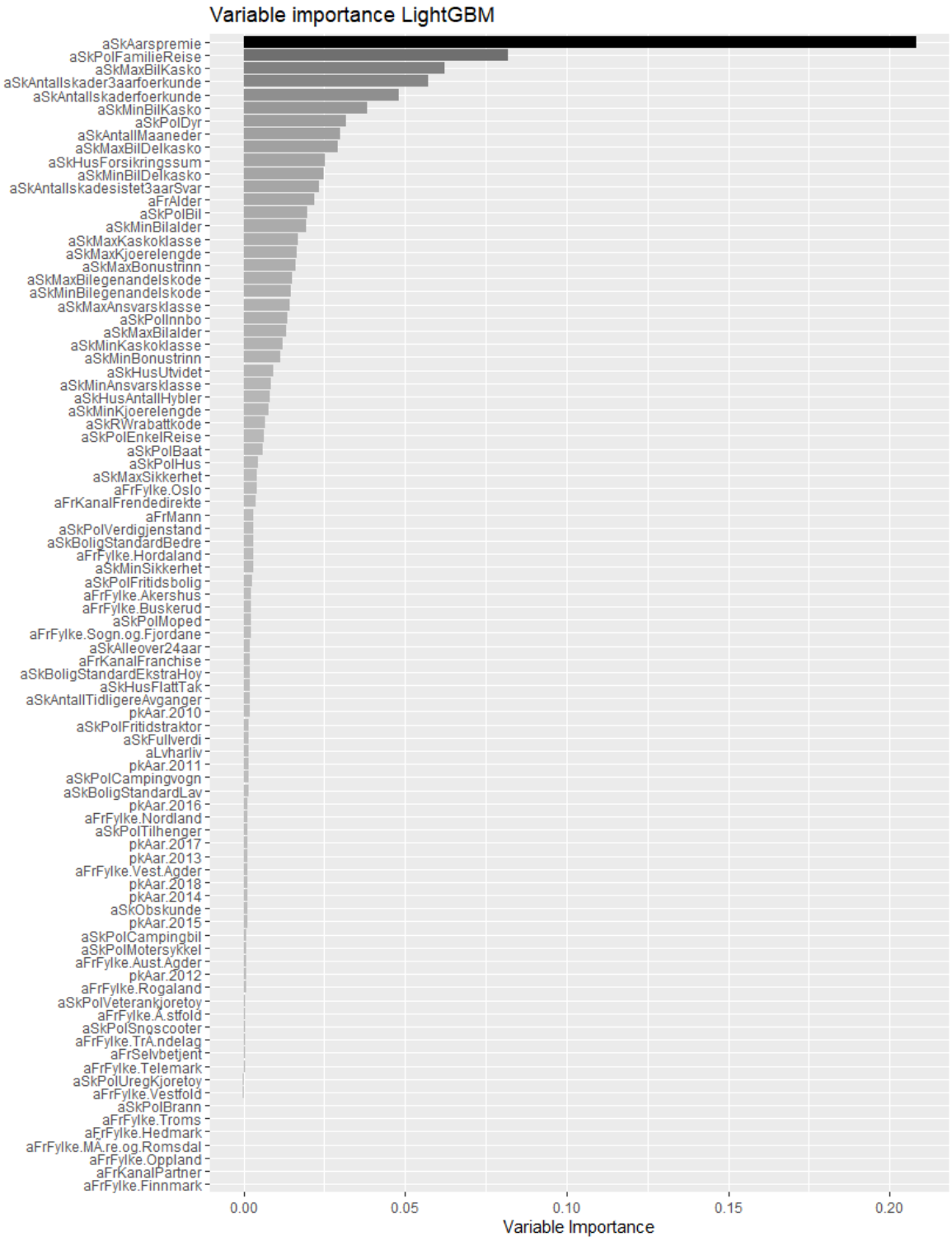

Figure 28 – Claim propensity variable importance using LightGBM

6 Modelling claim frequency

Modelling the claim frequency of customers can be useful to discover certain customer traits that distinguish customers with high claim frequencies from those with low claim frequencies. We will look further into the results achieved by our models, but also the customer traits which are most important to predict if a customer has a high claim frequency or not.

6.1 Model predictions

We have focused on two main methods to assess model performance, RMSE and MAE. We will measure the performance of the models using all explanatory variables, but also using only the yearly premium to see what additional value the individual customer data provides when modelling claim frequency. To understand the differences between models and discover the weak points of each model, we will provide some detailed statistics about the model responses.

6.1.1 RMSE comparison

In addition to the models used to determine the customer claim risk, we use simple averaging of the models to see how it compares to the more sophisticated stacked ensemble model. The overall results from the models using the yearly premium to those that use all variables indicate that there is value in using the individual customer data provided. The difference in performance, however, is smaller than it was when modelling claim propensity.

Model	RMSE (All variables)	RMSE (Yearly premium)
Neural Network	0.2650	0.2689
GLM	0.2657	0.2691
LightGBM	0.2640	0.2688
XGBoost	0.2642	0.2688
Random Forest	0.2658	0.2702
Stacked Ensemble	0.2626	0.2687
Averaging	0.2644	0.2688

Table 4 – Claim frequency RMSE comparison

Table 4 shows that the best performing model is the stacked ensemble with an RMSE of 0.2626. We can see it outperforms the best-performing singular model, LightGBM, with an RMSE of 0.2640. It is closely followed by the other gradient boosting model XGBoost, with an RMSE of 0.2642. Averaging of all algorithms gives us the fourth-best performing model with an RMSE of 0.2644, followed by the neural network achieving an RMSE of 0.2650. Our

benchmark GLM model achieves an RMSE of 0.2657, just beating out the Random Forest model, which again is the worst performer with an RMSE of 0.2658. We discovered that the GLM without penalty terms outperformed versions using ridge and lasso regularization when modelling claim frequency.

6.1.2 MAE comparison

Table 5 shows that there is a different order of the best-performing models when measuring performance with MAE.

Model	MAE (All variables)	MAE (Yearly premium)
Neural Network	0.1160	0.1253
GLM	0.1207	0.1236
LightGBM	0.1198	0.1230
XGBoost	0.1200	0.1253
Random Forest	0.1221	0.1235
Stacked Ensemble	0.1193	0.1241
Averaging	0.1218	0.1255

Table 5 – Claim frequency MAE comparison

The best performing model is the neural network with an MAE of 0.1160. The stacked ensemble is now the second-best performing model with an MAE of 0.1193, closely followed by the LightGBM model with an MAE of 0.1198. The XGBoost is now the fourth-best model with an MAE of 0.1200, followed by the benchmark model GLM with an MAE of 0.1207. Averaging achieves an MAE of 0.1218, and Random Forest again is the worst-performing model with an MAE of 0.1221. We see a significant gap between the models MAE using all variables to those using only yearly premium, similarly to when measuring with RMSE.

6.1.3 Differences in RMSE and MAE performance

As we can observe in the RMSE and MAE results for each model, there is a discrepancy in the order of the best-performing models between the two measures. While the neural network comfortably outperforms the other algorithms using the MAE measure, it is among the worst performers using RMSE. As described in section 3.5, RMSE penalizes large prediction errors harder than MAE, while MAE favors the average overall prediction errors to be low. By looking closer at the predictions made by our models in table 6, we can see why the discrepancy in performance exists.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Observed	0.0000	0.0000	0.0000	0.0784	0.0000	19.2678
XGBoost	-0.0217	0.0507	0.0704	0.0776	0.0940	3.5767
Neural Network	-0.1057	0.0545	0.0703	0.0784	0.0859	1.2846
LightGBM	-0.0277	0.0492	0.0701	0.0775	0.0954	0.9295
Random Forest	0.0239	0.0606	0.0734	0.0775	0.0892	0.7999
GLM	-0.1664	0.0504	0.0728	0.0774	0.0998	0.6125
Averaging	0.0157	0.0782	0.0824	0.0805	0.0849	1.3659
Stacked ensemble	0.0138	0.0774	0.0802	0.0784	0.0826	0.7101

Table 6 –Distribution of predicted claim frequency of all singular models

We see that the observed response is usually zero. This means that the customer does not make any claims. As the vast majority of customers do not have any claims in any given year, this means the model that predicts values closer to zero tends to have a lower MAE. The most conservative models are therefore rewarded with a low MAE, despite having more substantial prediction errors when measured by RMSE. We can illustrate how the RMSE and MAE behave by using a model predicting only one single value across the claim frequency range from 0 to 0.25 in figure 29. The red line indicates the performance of our respective best-performers using RMSE and MAE.

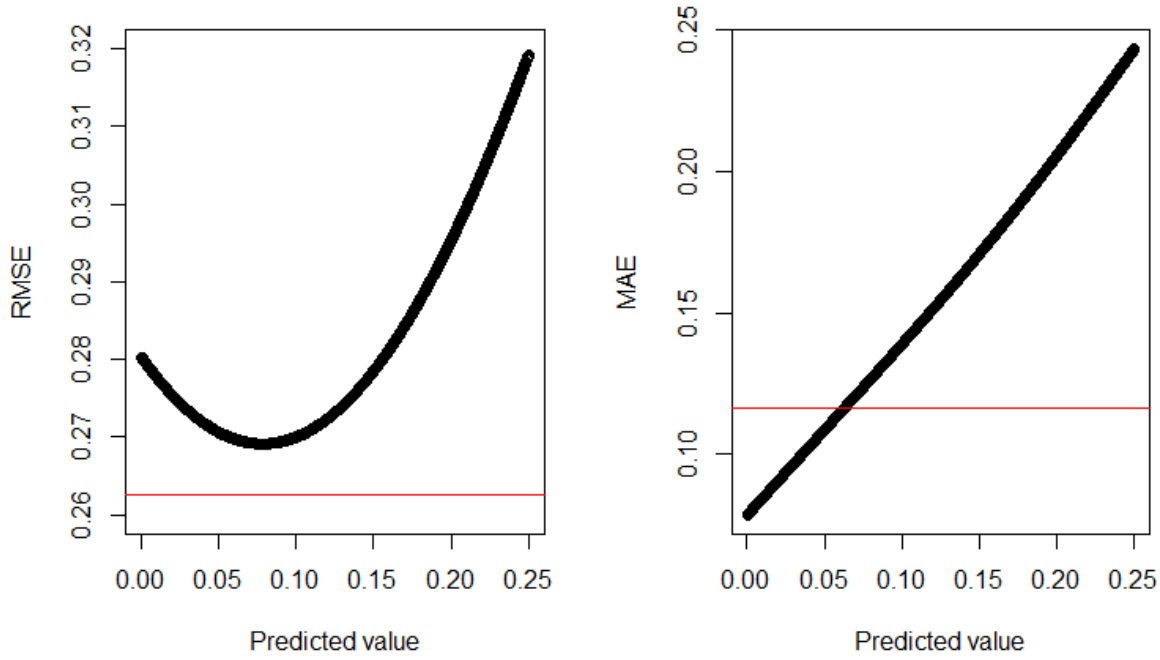

Figure 29 –RMSE and MAE plots showing error when predicting the same claim frequency on all observations

We can see that predicting zero on all observations will outperform all the other models using the MAE measure, with a score of 0.0783 compared to our best performing model, the neural

network, which achieved an MAE of 0.1160. Doing the same with RMSE will give us worse performance than all the other models. MAE, in other words, value models that predict closer to the most common outcome, rather than RMSE, which puts greater emphasis on outliers. As our main objective is to find out what customers present the most significant risk of high claim frequency, we consider the results of RMSE to be the more suitable estimate of model performance. The figure also illustrates that our stacked ensemble is able to distinguish between customers with different claim frequencies, as the optimal single claim frequency value of 0.0784 still performs considerably worse, with an RMSE of 0.2691 than all our other models with all explanatory variables available. Of our models using only the yearly premium to predict claim frequency, we can observe that the random forest model performs worse than consistently predicting 0.0784, and the GLM performs equally. All the other models are still able to outperform the optimized single value prediction. This illustrates that these models are not particularly useful using only the yearly premium as an explanatory variable, but also that the other models are able to distinguish between the claim frequency of customers based upon the individual customer data at our disposal.

An approach which is not part of the scope of this thesis, but could possibly improve the prediction in view of the above result, is to perform a two-step conditional prediction routine; First, one uses the customer claim risk model as in Section 5 to predict whether the customer will have one or more claims. Secondly, one could repeat the above model fitting but using data where all individuals have one or more claims. This model can then be used to predict the actual number of claims for an individual where the customer claim risk model has predicted one or more claims. Conversely, if the customer claim risk model predicts zero claims, then one simply let the predicted number of claims to be equal to zero.

6.2 Effects of explanatory variables on customer claim frequency

Figure 30 shows that the random forest model assigns the greatest importance to the yearly premium, age, previous customer history, and car insurance policy related variables.

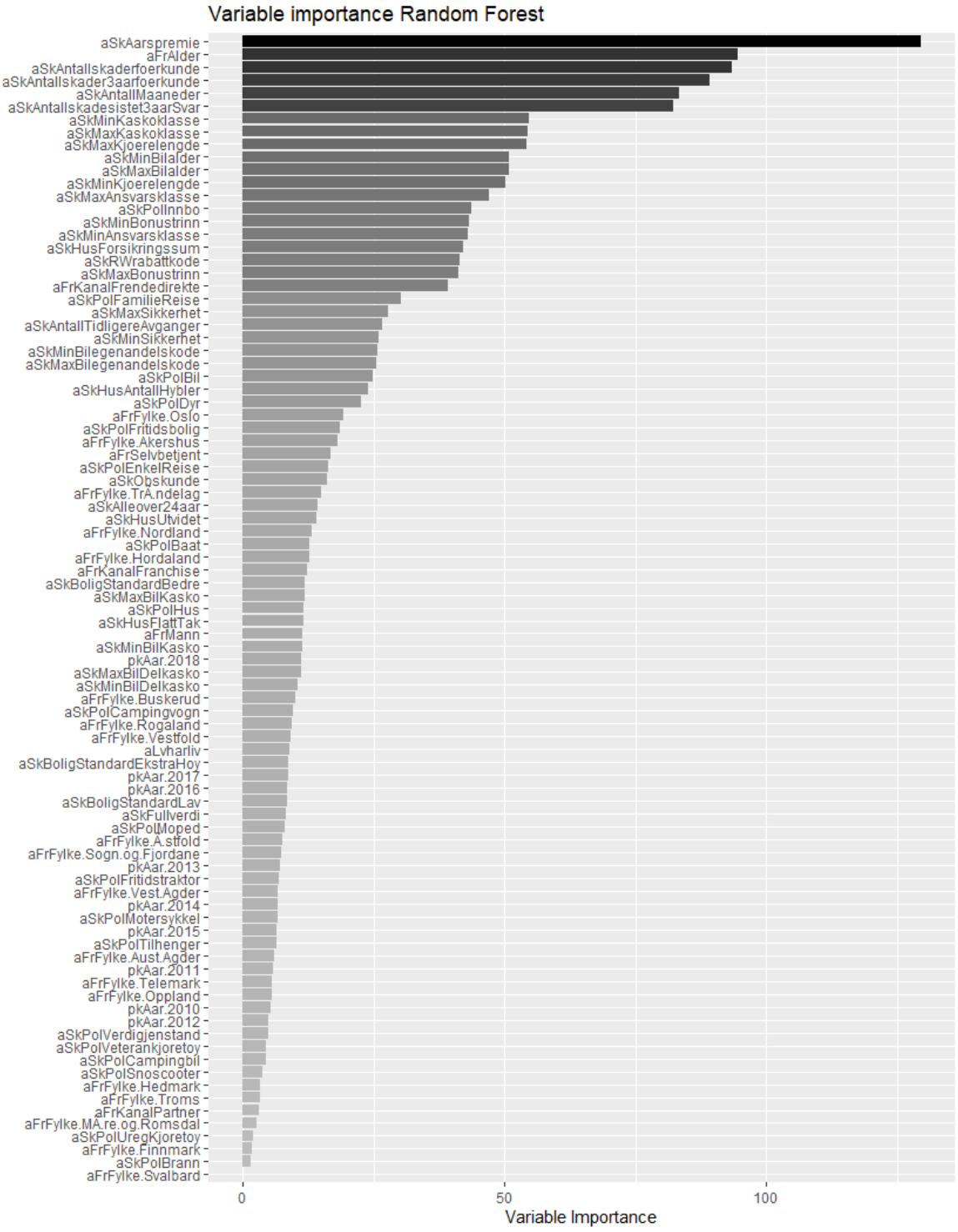

Figure 30 – Claim frequency variable importance using random forest

Figure 31 shows the variable importance of our best-performing model. It shares certain variables with the Random Forest importance, such as yearly premium amount, previous customer history, car insurance variables, and car insurance policy. Unlike the random forest model, it puts more emphasis, relatively, on the house insurance policy and variables.

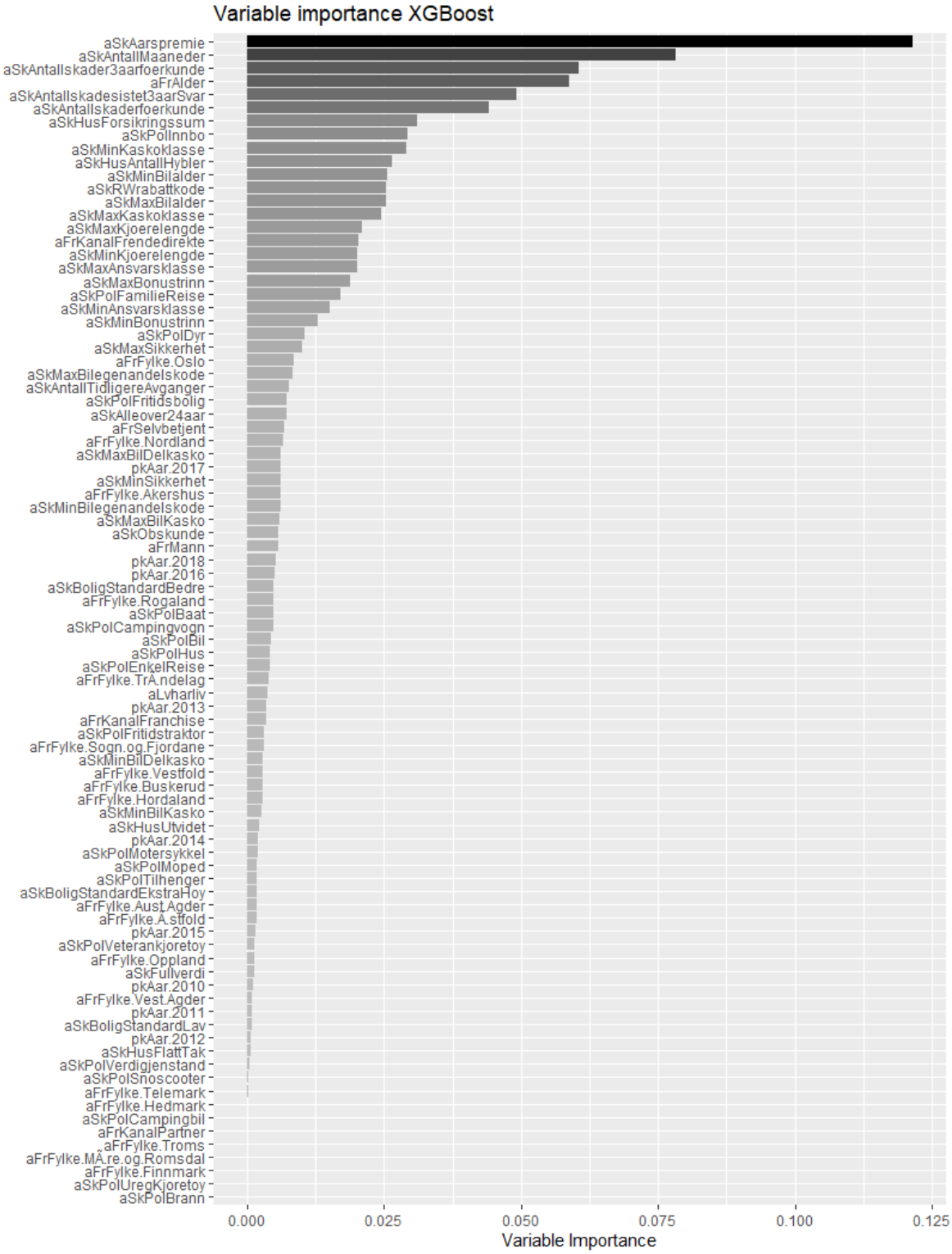

Figure 31 – Claim frequency variable importance using XGBoost

Just like when modelling claim propensity, the LightGBM model still assigns similar variable importance to the XGBoost model. Figure 32 does however, show that there are various differences in the importance order. LightGBM puts more emphasis on customer relationship length, car insurance discount, and pet insurance.

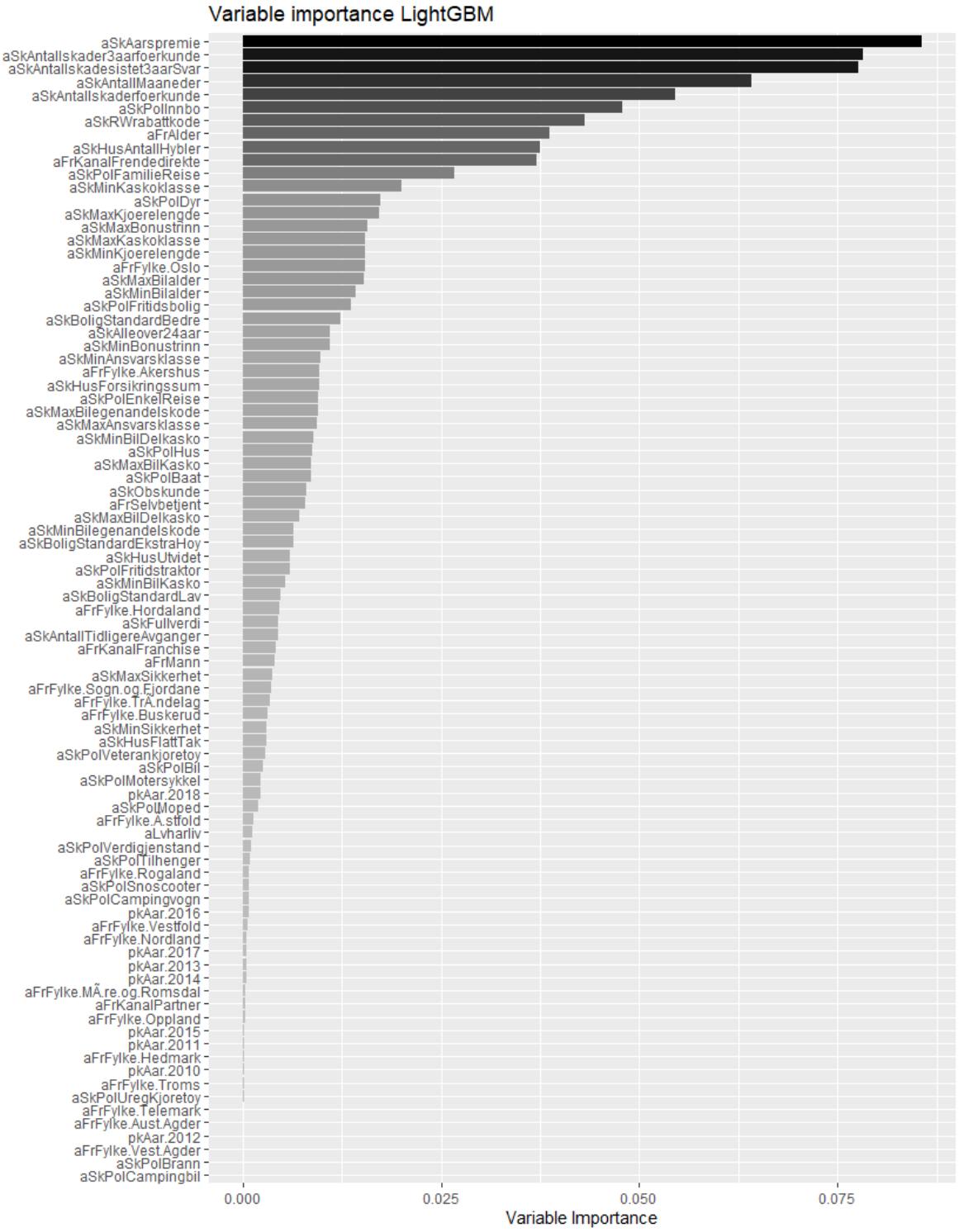

Figure 32 – Claim frequency variable importance using LightGBM

7 Modelling total claims

In addition to determining the customer claim risk and customer claim frequency, we want to model the customer's total claim. It consists of the total claims made a given year by each customer. It is a stochastic variable, which is especially hard to model as it consists of two different stochastic variables, claim frequency and claim severity. In our dataset, we do not have access to the severity of individual claims, but we can find the average claim size using the claim frequency and the total claim size.

7.1 Model predictions comparison

Similar to when modelling claim frequency, we have focused on RMSE and MAE to assess model performance. We will also distinguish between the different models with all available data, and those using only the yearly premium, to see if the individual customer data provides any additional value when predicting total claim size. To understand the differences between models and discover their weak points, we will provide some detailed statistics about the model responses.

7.1.1 RMSE comparison

Table 7 shows that using the yearly premium to predict total claims is almost as effective as using all available data. While this shows that the individual customer data is less important to determine total claims size than claim propensity and frequency, it underlines the importance of choosing the correct model to achieve good results, as some models are able to outperform the others using only yearly premium as an explanatory variable. There are multiple reasons why this might be the case. The well-proven individual policy models used by Frende have access to more data and have been developed specifically to leverage that data and the company insights. The high degree of randomness in total claim size also seems to put more emphasis on correctly distributed models than our other responses.

Model	RMSE (All variables)	RMSE (Yearly premium)
Neural Network	76 775.35	76 806.90
GLM	76 773.84	76 806.72
LightGBM	76 787.61	76 812.96
XGBoost	76 849.83	76 956.98
Random Forest	76 791.00	77 819.30
Stacked Ensemble	76 802.38	76 830.53
Averaging	76 781.37	76 865.87

Table 7 – RMSE comparison of total claim predictions

Table 7 also shows that the best performing model is the GLM, with an RMSE of 76 773.84. The best performing GLM was regularized with a combination of lasso and ridge, but the gains from a regular GLM was modest. The differences between the models, when measured by RMSE, are quite small because of a handful of massive claims which the models are not able to predict. The neural network achieves an RMSE of 76 775.35, while the LightGBM model is the third-best performer with an RMSE of 76 787.61. Averaging the models gives us an RMSE of 76 781.37, while the random forest performs well compared to its performance on the previous response variables, and is fifth-best with an RMSE of 76 791.00. The stacked ensemble and XGBoost perform quite poorly compared to what they achieved at claim risk and frequency with an RMSE of 76 802.38 and 76 849.83, respectively.

7.1.2 MAE comparison

Model	MAE (All variables)	MAE (Yearly premium)
Neural Network	9 545.81	9 697.52
GLM	9 733.84	9 719.53
LightGBM	9 602.85	9 523.99
XGBoost	8 836.22	9 517.81
Random Forest	9 740.47	9 944.35
Stacked Ensemble	8 866.14	10 799.14
Averaging	9 454.43	9 349.66

Table 8 – MAE comparison of total claim predictions

Table 8 shows that similarly to when modelling claim frequency, there is a different order in the best-performing models when measuring performance with MAE. The ranking order, however, is quite different. The best performing model is the now the XGBoost model with an MAE of 8 836.22. The stacked ensemble is now the second-best performing model with an MAE of 8 866.14. There is a large gap to the third-best performing model measured, averaging, having an MAE of 9 454.43. The neural network achieves an MAE of 9 545.81, and LightGBM follows with an MAE of 9 602.85. The large difference between the relatively similar LightGBM and XGBoost models can be considered slightly surprising, being closely matched predicting the other response variables. Our GLM, with a combination of lasso and ridge regularization, achieves an MAE of 9 733.84, just beating out the random forest model MAE of 9 740.47. An interesting observation is that several models perform better using only the yearly premium as an explanatory variable, which was not the case when measuring by RMSE.

7.1.3 Differences in RMSE and MAE performance

When comparing the RMSE and MAE of the models, we can see that the models that perform well using RMSE, often performs among the worst using MAE, and the other way around. There seems to be a tradeoff between achieving a good RMSE and MAE. The results from claim frequency seemed to have a similar tradeoff, but not to the degree seen in the total claim size modelling. This results in the worst-performing model measured by RMSE, the XGBoost, achieving the best results measured by MAE.

We described a possible cause of discrepancies between RMSE and MAE performance while presenting the claim frequency results. We can see the same patterns when modelling the total claim size, albeit slightly enhanced. Table 9 provides a detailed overview of the model predictions and gives us an indication of why the discrepancy in performance exists. We can see that the XGBoost consistently predicts low values but have some extreme predictions when compared to the others. Given the characteristics of RMSE and MAE, this means it will have a low average error but will be punished hard for its highest predictions if the customer has zero claims according to the RMSE measure.

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Observed	0.00	0.00	0.00	6 117.00	0.00	17 136 218.00
XGBoost	392.10	1 769.60	4 114.50	4 868.40	6 807.60	338 102.20
Neural Network	15.62	1 665.30	4 890.22	5 916.24	8 677.04	60 086.64
LightGBM	1 076.00	2 404.00	4 898.00	5 968.00	8 252.00	96 391.00
Random Forest	795.00	2 586.00	5 328.00	6 134.00	8 670.00	85 475.00
GLM	-7 205.00	1 954.00	5 089.00	6 123.00	9 030.00	73 764.00
Averaging	-126.20	2 152.70	4 883.20	5 801.00	8 254.80	257 083.70

Table 9 – Distribution of predicted and observed total claim size of all singular models

Similarly to the claim frequency models, we can outperform the MAE measures of the models by predicting zero total claim across all observations, resulting in a score of 6 117.18. When predicting zero on all responses while measuring with RMSE, we achieve a score of 77 193.93, significantly worse than our worst-performing model. We can also observe that the GLM has negative predictions, and as negative claims are not to be expected, this means there are further gains to be made by left-truncating the predictions. In tables 10 and 11, we can observe the changes in RMSE and MAE, respectively, caused by truncating the predictions.

Truncating	XGBoost	Neural Network	LightGBM	Random Forest	GLM	Averaging
<0	76 847.15	76 777.99	76 787.61	76 791.00	76 772.99	76 781.57
< 1 000	76 847.56	76 778.48	76 787.61	76 790.94	76 773.06	76 781.57
< 5 000	76 885.00	76 800.71	76 810.09	76 811.00	76 794.44	76 804.30

Table 10 – RMSE of left-truncated total claim size predictions

Truncating	XGBoost	Neural Network	LightGBM	Random Forest	GLM	Averaging
<0	8 852.19	9 517.55	9 602.85	9 740.47	9 669.91	9 447.24
< 1 000	8 752.11	9 462.82	9 602.85	9 685.45	9 640.56	9 382.44
< 5 000	7 816.32	8 690.04	8 600.92	8 818.15	8 912.01	8 520.46

Table 11 – MAE of left-truncated total claim size predictions

We can also illustrate how the RMSE and MAE behave when using a model predicting only one value across the range from 0 to 25 000 in figure 33. The red line indicates the performance of our best singular model on RMSE and MAE, respectively.

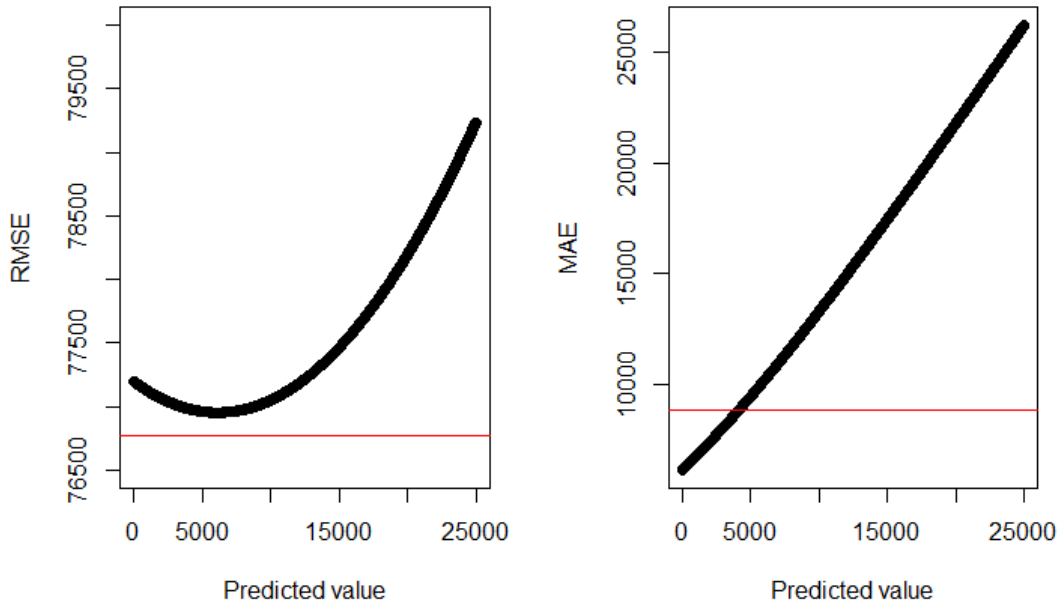

Figure 33 – RMSE and MAE plots showing error when predicting the same total claim on all observations

It shows how it is possible to beat our MAE scores by simply predicting anything from zero to 4 157 for every observation. It also shows that predicting any single value on all observations will result in worse RMSE performance, which underlines why it has been our preferred model performance metric. By out-performing the single value that optimizes RMSE in our test set, it also shows that despite the significant occurrence of randomness in which customers have high total claims, our models are still able to distinguish customers likely to

have high total claims. While the difference might not seem large, this is mainly caused by some very high claims severely increasing the RMSE. The large discrepancy between the RMSE and MAE illustrates this point as the squared residuals of RMSE makes it around eight times larger than the MAE on average across our models.

Similarly to our suggestions in the claim frequency result section, we could possibly model total claim size by using a three-step conditional prediction routine, by adding the customer claim risk model as described in section 5 to predict whether the customer will have one or more claims, then model the claim frequency of the customer and finally the total claim size. With this approach, we let the total claim size be a function of our existing models used in sections 5, 6, and 7. This model could be used to predict the total claim size given the claim frequency and claim risk of the individual customer.

7.2 Effects of explanatory variables on customer claim

We can observe in figure 34 that house insurance sum is now the second-most important explanatory variable in our dataset, and that previous customer history is less important to determine total customer claim when compared to claim frequency. This is likely related to the biggest customer claims often being in relation to housing policies. Housing insurance, inventory and other housing-related claims tend to be rare occurrences but result in more severe claims than the other policies offered by Frende. Similar to claim propensity and frequency, the car insurance explanatory variables are important to the model.

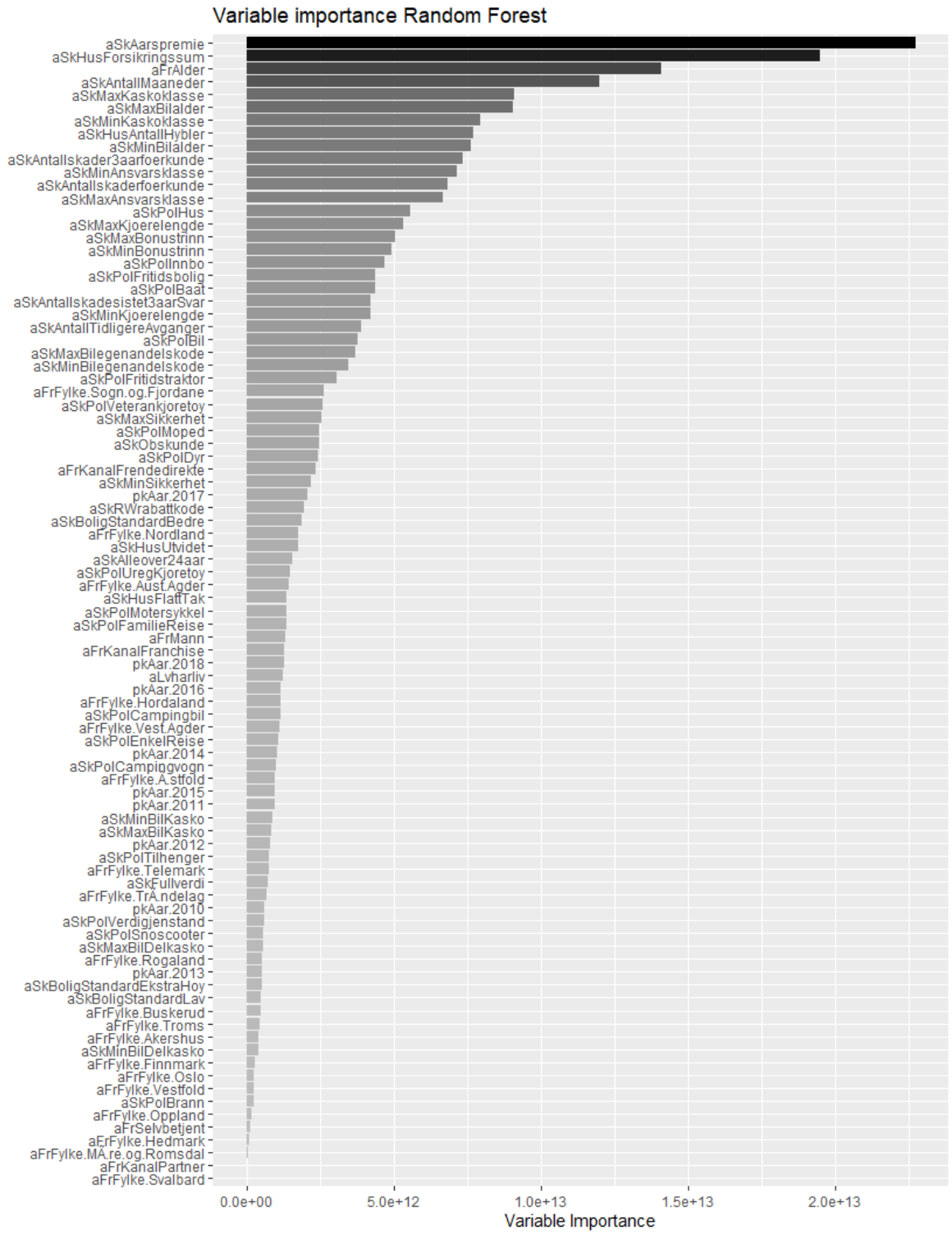

Figure 34 – Total claim variable importance using random forest

From the XGBoost variance importance shown in figure 35, we can see some of the same patterns observed in the random forest plot. We can see that the house insurance sum is the second-most important explanatory variable after the yearly insurance premium. Otherwise, the most important explanatory variables are similarly ranked, with some notable counties and years carrying more significance for the results obtained by the XGBoost model.

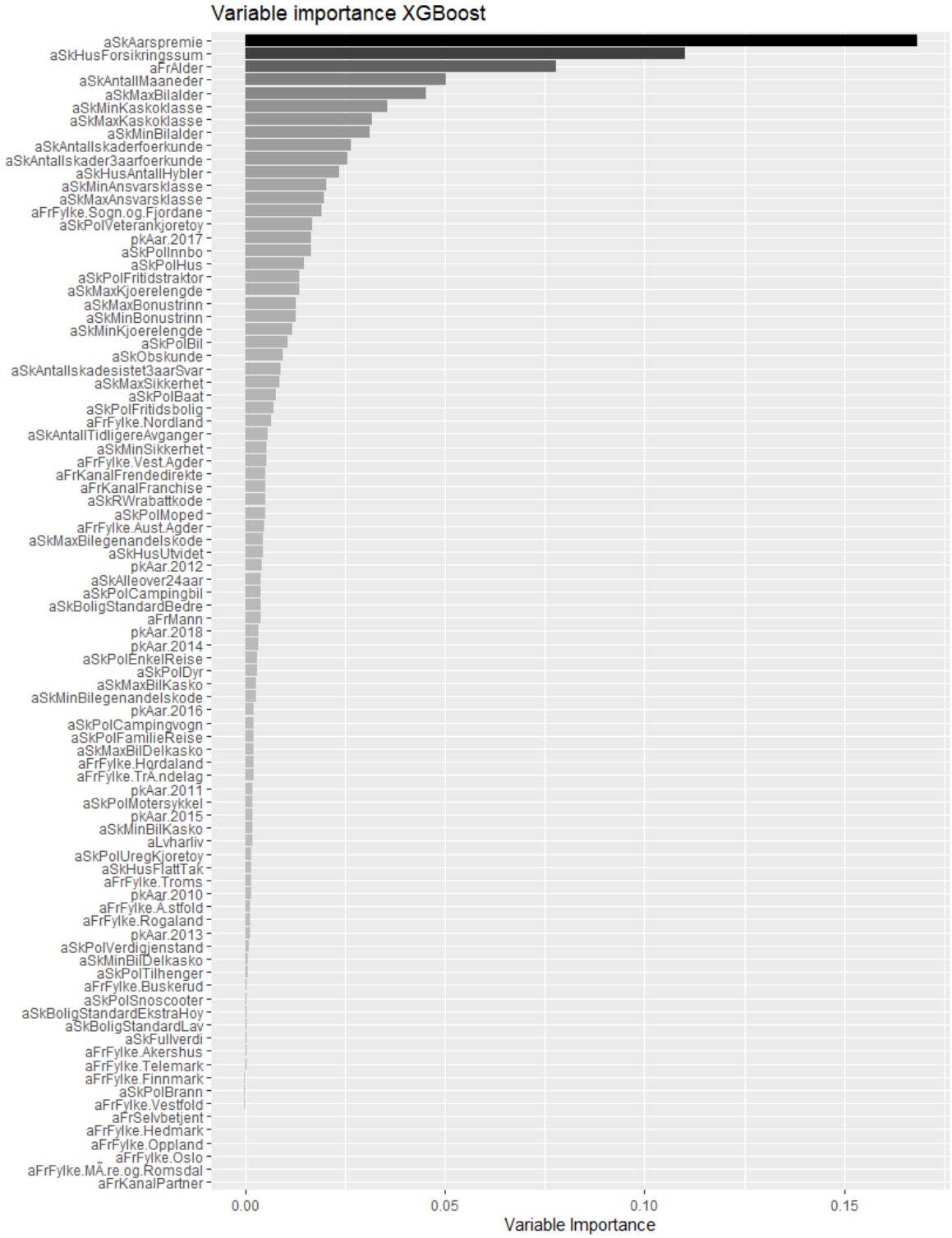

Figure 35 – Total claim variable importance using XGBoost

We can observe in figure 36 that LightGBM has some differences to the XGBoost model. It puts greater emphasis on customer history than the XGBoost and third-party liability insurance. Unlike the XGBoost, the LightGBM ranks the categorical variables such as counties and year as less important. The common theme among the model's variable

importance is car insurance variables, house insurance policy variables, age, yearly customer insurance premium, and customer history variables.

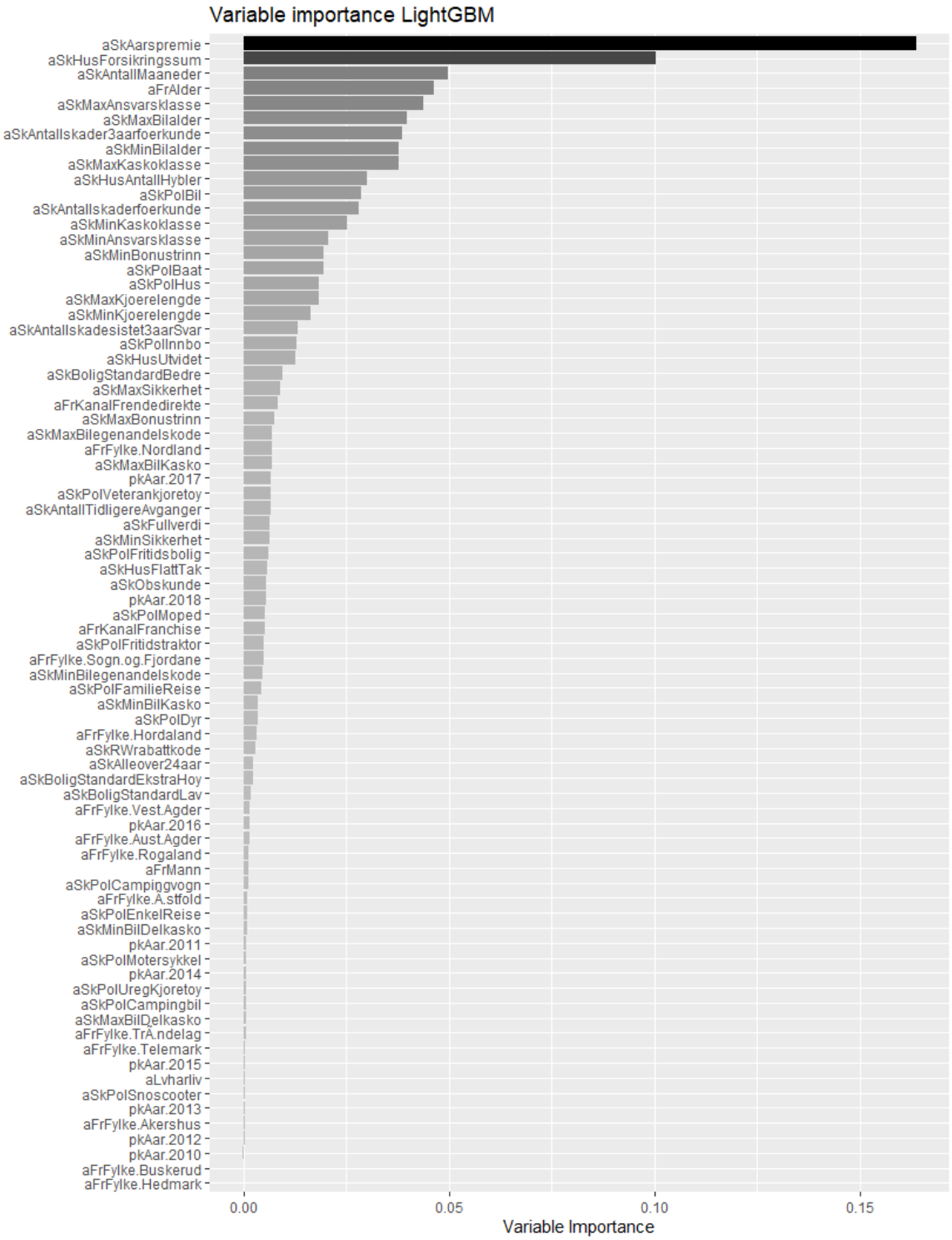

Figure 36 – Total claim variable importance using LightGBM

8 Discussion and conclusion

8.1 Discussion

When modelling claim propensity, we found that we were able to leverage the individual data provided to increase the model performance significantly when compared to using the yearly insurance premium. Claim propensity is the response variable that seems to make the most use of the individual data, which was consistent with what we expected. In our claim frequency modelling, we observed many of the same models performing well, with the stacked ensemble seemingly even more useful than when modelling claim propensity. When it came to the total claim size, we found that the choice of model was more important than added explanatory variables. Using the full set of explanatory variables yielded solid improvements across claim propensity, claim frequency, and to a lesser degree, total claim size. While we expected both claim propensity and claim frequency to have considerable improvements by including all variables, the slight improvements gained in total claim size predictions could also be of value. Overall, we found these results to be encouraging when it comes to modelling individual customer risk across multiple policies.

There are some interesting findings in the model performance across claim propensity, claim frequency, and total claim size. Our findings show that the gradient boosting models LightGBM and XGBoost perform particularly well when modelling claim propensity and frequency. These models are quite popular in data science competitions and are often included in the winning solutions, which is understandable given our findings. We do see that the most sophisticated models might not result in better predictions across all responses. We found that predicting total claim was particularly tricky when trying to optimize singular gradient boosting and neural network models. This could be caused by the high degree of randomness associated with total claim size, and that the models struggle to distinguish customers with large claim sizes, in contrast to their claim propensity and frequency. When compared to the GLM models, which not only are easy to implement, adaptable, and perform consistently while modelling all three responses, we saw that our most flexible models were not able to beat our benchmark GLM, when measured by RMSE.

This underlines an important aspect when applying flexible machine learning models such as LightGBM, XGBoost, and the neural network to predict customer risk. They are very sensitive to hyperparameter optimization, and although we have spent considerable time

optimizing these hyperparameters, we are confident that there are further gains to be made. Without time and computational constraints, we would expect these models to have extra room for improvement across all response variables. This underlines the most important strength, but also the greatest weakness of these models. They are highly flexible and do a great job fitting the data, and while the flexibility means these models have the potential to outperform traditional statistical learning models, it also makes them prone to overfitting. They all have built-in features to avoid overfitting, and many of these features depend on hyperparameter optimization. If these hyperparameters are not correctly set, their performance significantly decreases. Working with our thesis, we have seen large differences in performance using the default hyperparameters and the optimized hyperparameters found through grid search and bayesian optimization. Finding the correct hyperparameters are often highly dependent on the dataset, the response variable, and varies between different purposes and applications. This makes hyperparameter optimizing time-consuming, computationally intensive, and difficult for users without domain knowledge.

It is also important to note the strengths and overall impressive performance of the GLM model. While there have been introduced many new and sophisticated approaches to take advantage of the increased computing power and data richness available, the GLM model still performs relatively well across all response variables. It has served as an excellent benchmark, and was the strongest performer predicting total claim size. The lack of flexibility makes it lack the peak performance of the other machine learning models, but the stability and robustness of the model are important features in many applications. We also discovered that due to the high-dimensional nature of our dataset, there were gains to be made by using lasso and ridge regularization to the GLM. While the claim frequency did not see any improvements using regularization terms in the GLM, we saw a slight improvement for total claim size and a considerable improvement when modelling the claim propensity.

Perhaps the most interesting discovery from our results is that ensemble methods were able to improve the performance of our best models. While gradient boosting and bagging have become more widespread through popular models such as XGBoost and Random Forest, we have shown how using stacked ensembles can further improve predictions consisting of already well-performing models. The stacked ensemble can leverage the law of large numbers to enhance the performance of our singular models, and it performed better than all the individual models used to assess claim propensity and frequency. The improvements in results

compared to the individual models could be observed using all variables, but also when we only used the yearly premium, which reveals its stability and robustness as another strength. Combining multiple flexible models and less flexible models such as GLMs in a stacked ensemble makes it possible to achieve increased performance without sacrificing stability.

It is important to point out that there is "no such thing as a free lunch", and stacked ensembles come with some downsides. They are computationally intensive to train, and by combining multiple models, they will spend more time predicting results. The added complexity also adds to a problem found across all our complex models, the black-box issue. XGBoost, LightGBM, and neural networks are significantly less transparent about how it arrives at its predictions than decision trees and GLMs. Creating large ensembles of multiple complex models only adds to the problem. While it is possible to uncover the variable importance of the meta-model and then all its underlying models, it will be hard to explain the decision-making process in understandable terms for an end-user. Depending on the specific use case, this could either result in a regulatory issue or just a negligible inconvenience.

8.2 Conclusion

In our thesis, we have investigated the value of using individual customer data to model customer claims, claim frequency, and total claims. Traditionally, the insurance industry has focused on modelling individual policies using conservative statistical modelling techniques. By using individual customer data across multiple policies, we could observe prediction improvements across all three responses. Our thesis has also shown promise in using modern machine learning model to leverage this individual data in the best possible way.

In addition to highlighting the potential of analysing customers as a whole, rather than its individual insurance policies, our thesis also highlights the importance of choosing the correct model and the benefits associated with ensemble techniques. We were able to beat our benchmark GLM models in two out of our three responses, using various individual models and ensembles. We found differences in overall performance between these individual models, but also methods to combine these models to enhance overall performance and stability. Among our singular models, we found the gradient boosting models LightGBM and XGBoost to be particularly strong performers. The most impressive performer was a stacked ensemble leveraging multiple gradient boosting models, random forests, neural networks, and generalized linear models. It was our best-performing model modelling claim propensity and

frequency. It did not only provide performance benefits but also proved more robust on subsamples and across different features than our other models. We did, however, find that GLMs are widespread in the insurance industry for good reason, as it was our best performer modelling the total claim size. While it is difficult to evaluate the benefits of risk assessing customers across multiple policies in monetary terms, there are reasons to believe that added insights in this area could be valuable for Frende. It could make it easier to avoid particularly risky customers, but also make it easier to retain attractive customers by offering discounts based on a customer's entire policy portfolio. While multiple product discounts are already common in the insurance industry, better estimates on customer risk could help to distinguish which customers should be offered these discounts, and how large each discount should be.

8.3 Suggestions for further studies

Within insurance claim modelling, there are two different presumptions about claim size and claim frequency, one of which assumes that they are independent - and the other assumes that they are dependent variables. If we presume dependence, we hypothesized that models including predictions about both claim size and frequency could potentially deliver good results. As we found models predicting claim propensity to perform well, we also decided to try out a three-stage model, including claim propensity, claim frequency, and total claim size. By basing the predictions on several models, it might also increase the stability of predictions. However, as we pursued this task, we quickly found this approach to be very time-consuming. The high dimensionality of the data and multiple models used both sequentially and in parallel makes hyperparameter optimization time-consuming and adds complexity overall. We ran trials using smaller subsets of our provided dataset and found that any increase in performance seemed to be modest when compared to the time spent adjusting parameters and selecting features. Despite these added complexities, we find it to be an interesting concept that possibly could provide value when modelling claims.

An alternative that might be less computationally expensive but still able to make use of the same characteristics, is a stacked ensemble using predictions of claim propensity, claim frequency, and claim size as inputs to a meta-model predicting total claim size. Such a model would massively decrease data dimensionality, without losing too much information in the process. It would also allow leveraging the entire training set for model training purposes.

8.4 Disclaimers, sources of error and limitations

In an empirical study, it is important to disclose possible sources of error and any limitations the results and conclusions might have. We have written a step-by-step overview:

- As our data is provided by a third party, we cannot exclude the possibility of data errors. There might be some errors from manual registering or transfers between internal systems.

- Programming errors in R cannot be ruled out, given the vast amounts of code required to run our models. We have repeatedly checked for errors, made sure results seem reasonable, and in some cases, we ran simple small-scale experiments on our models to ensure they behave as expected. We have also gone through our code to make sure that there are no data leakages. By data leakages, we mean data not supposed to be seen by the model at the training stage, so that it has information that could affect results when predicting on the unseen test data.

- Our dataset is recent, so our findings should be relevant for modelling current insurance customer risk. There is, however, an unbalanced number of observations for each year. As described in our data section, Frende has experienced substantial growth in their customer base, and therefore have relatively few customers in 2010 compared to 2018. The customer growth might make it more difficult for our models to account for any important changes which could have occurred in customer behaviour during the period. One example of this is the increasing entrance of electric cars, which have introduced changes to the car insurance policies in the later period of our dataset. These changes could potentially skew model results when applied to new data.

- Even though the dataset contains many observations, it is important to keep in mind that Frende makes up 3.5 % of the Norwegian insurance market in 2019 (Slettemoen, 2020). Their customer base is also quite skewed geographically with their market share in west-coast and southern counties making up anywhere between 10-15 %, compared to their national average of 3.5 %. We still believe that insurance companies and their customers are homogenous enough to make our results universally applicable. It is still important to consider that the Norwegian insurance customers will likely have specific characteristics, potentially making the source of error larger in an international context.

9 Literature

Aggiwal, R. (2017). Introduction to XGBoost.
Retrieved from: https://dimensionless.in/introduction-to-xgboost/

Ash, R. B. (2008). *Basic probability theory*. Courier Corporation.

Beale, H. D., Demuth, H. B., & Hagan, M. T. (1996). Neural network design. *Pws, Boston*.

Boland, P. J. (1989). Majority systems and the Condorcet jury theorem.
Journal of the Royal Statistical Society: Series D (The Statistician), *38*(3), 181-189.

Borovykh, A., Bohte, S., & Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691*.

Breiman, L. (1997). *Arcing the edge*.
Technical Report 486, Statistics Department, University of California at Berkeley.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5-32.

Breiman, L., & Spector, P. (1992). Submodel selection and evaluation in regression.
The X-random case. *International statistical review/revue internationale de Statistique*, 291-319.

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

Brown, C. D., & Davis, H. T. (2006). Receiver operating characteristics curves and related decision measures: A tutorial. *Chemometrics and Intelligent Laboratory Systems*, *80*(1), 24-38.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific model development*, *7*(3), 1247-1250.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Chen, Y. Y., Lin, Y. H., Kung, C. C., Chung, M. H., & Yen, I. (2019). Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes. *Sensors*, *19*(9), 2047.

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*(1), 6.

Claesen, M., & De Moor, B. (2015). Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*.

Det kongelige finansdepartement. (2013–2014). Prop. 87 L. Endringer i børsloven og verdipapirhandelloven mv. Og lov om kredittvurderingsbyråer. Retrieved from: https://www.regjeringen.no/contentassets/528d8efa0fc1405a91d1be5d182b77ed/no/pdfs/prp201320140087000dddpdfs.pdf

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of statistics*, *32*(2), 407-499.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), 861-874.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1, No. 10). New York: Springer series in statistics.

García, V., Mollineda, R. A., & Sanchez, J. S. (2010, August). Theoretical analysis of a performance measure for imbalanced data. In *2010 20th International Conference on Pattern Recognition* (pp. 617-620). IEEE.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Greenwell, B., Boehmke, B. & Cunningham, J. (2019, Jan). Package 'gbm'. Retrieved from: https://cran.r-project.org/web/packages/gbm/gbm.pdf

Hand, D. J., & Till, R. J. (2001). A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine learning*, *45*(2), 171-186.

Harris, D., & Harris, S. (2010). *Digital design and computer architecture*. Morgan Kaufmann.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, *20*(8), 832-844.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, pp. 3-7). New York: springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An introduction to statistical learning* (Vol. 112, pp. 3-7). New York: springer. Corrected edititon.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems* (pp. 3146-3154).

Kuncheva, L. I. (2014). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

Liu, Y. (2002). The evaluation of classification models for credit scoring. *Institut für Wirtschaftsinformatik, Georg-August-Universitat Göttingen*.

Lobo, J. M., Jiménez-Valverde, A., & Real, R. (2008). AUC: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography*, *17*(2), 145-151.

Lin, G., Sun, N., Nepal, S., Zhang, J., Xiang, Y., & Hassan, H. (2017). Statistical twitter spam detection demystified: performance, stability and scalability. IEEE access, 5, 11142-11154.

Mahto, K. (2019). One-Hot-Encoding, Multicollinearity and the Dummy Variable Trap. Retrieved from: https://towardsdatascience.com/one-hot-encoding-multicollinearity-and-the-dummy-variable-trap-b5840be3c41a

MathWorks. (2016). Detector Performance Analysis Using ROC Curves. Retrieved from: mathworks.com/help/phased/examples/detector-performance-analysis-using-roc-curves.html

McLachlan, G. J., Do, K. A., & Ambroise, C. (2005). *Analyzing microarray gene expression data* (Vol. 422). John Wiley & Sons.

Mitchell, R., Adinets, A., Rao, T., & Frank, E. (2018). Xgboost: Scalable GPU accelerated learning. *arXiv preprint arXiv:1806.11248*.

Molinaro, A. M., Simon, R., & Pfeiffer, R. M. (2005). Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, *21*(15), 3301-3307.

Mueller, J. P., & Massaron, L. (2016). *Machine learning for dummies*. John Wiley & Sons.

Nelder, J.A. & Wedderburn, R.W.M. (1972). Generalized Linear Models. Reviewed work(s): Journal of the Royal Statistical Society. Series A (General), Vol. 135, No. 3, (p.370-384).

Norvig, P. R., & Intelligence, S. A. (2002). *A modern approach*. Prentice Hall.

Pan, R., & Yang, T. (2011). *A glm approach to optimal alt test plans for weibull distribution with type-I censoring* (No. 2011-01-0799). SAE Technical Paper.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013, February). On the difficulty of training recurrent neural networks. In *International conference on machine learning* (pp. 1310-1318).

PennState Eberly College of Science. (2020, March). Introduction to Generalized Linear. Models. Retrieved from: https://online.stat.psu.edu/stat504/node/216/

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, *6*(3), 21-45.

Pontius, R. G., & Parmentier, B. (2014). Recommendations for using the relative operating characteristic (ROC). *Landscape Ecology*, *29*(3), 367-382.

Ripley, B. D. (2007). *Pattern recognition and neural networks*. Cambridge university press.

Rokach, L., & Maimon, O. (2005). Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *35*(4), 476-487.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85-117.

Slettemoen, H.T. (2020, Jan). Snuoperasjon ga historisk resultat for Frende. Retriever from: https://www.frende.no/aktuelt/historisk-resultat/

Swets, J. A. (2014). *Signal detection theory and ROC analysis in psychology and diagnostics: Collected papers*. Psychology Press.

Tishby, N., & Zaslavsky, N. (2015, April). Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)* (pp. 1-5). IEEE.

Van der Laan, M. J., Polley, E. C., & Hubbard, A. E. (2007). Super Learner. Statistical Applications of Genetics and Molecular Biology, 6, article 25.

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research*, *30*(1), 79-82.

Zhang, Z., Zhao, Y., Canes, A., Steinberg, D., & Lyashevska, O. (2019). Predictive analytics with gradient boosting in clinical medicine. *Annals of translational medicine*, *7*(7).