Norwegian School of Economics
Bergen, Spring 2022

The Impact of Machine Learning and Aggregated Data on Corporate Insurance Modelling

An Empirical Study on the Prospective Gains of Machine Learning Techniques Using New Data Sources In the Insurance Industry

Tonje Hellestøl & Petter Eriksen

Supervisor: Lars Jonas Andersson

Master thesis, MSc in Economics and Business Administration

Major: Financial Economics & Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

This thesis was written as a part of the Master of Science in Economics and Business Administration at NHH. Please note that neither the institution nor the examiners are responsible – through the approval of this thesis – for the theories and methods used, or results and conclusions drawn in this work.

Abstract

This thesis investigates the potential applicability of machine learning techniques in predictive modelling on corporate insurance customers. The focus is on predicting a binary classification of claim occurrences and a customer's total claim size. Additionally, to illustrate practical usage, the respective best performing models were combined in an experimental setting to predict total expected cost and to identify good customers.

The data set is supplied by Frende Forsikring and consist of aggregated customer data. The aggregated data summarizes a company's characteristics, total premiums, number of claims, claim sizes and the policies they hold. Prior to data preprocessing the data consist of 26 293 different companies totaling 116 219 observations and 436 variables.

The study is split in two. First, the machine learning techniques CART, Random Forest, XGBoost and Neural Networks are compared with a benchmark GLM. Secondly, the thesis explores the predictive gain of aggregated data by using three input groups: the premium, using the initial aggregated data and using aggregated data with feature engineered time variables.

The results show that all machine learning models outperformed GLM when classifying claim occurrences. Additionally, all models showed an increase in predictive capabilities when including aggregated data, but little to no gain including time variables. XGBoost was the best performing model with an ROC-AUC of 0.8457. Resampling techniques did not contribute significantly to the performance to any of the models. In terms of predicting total claim size, no models produced satisfactory results. XGBoost performed best with a RMSE of 271725. The majority of the models performed best with premium as the only feature, indicating that the usage of aggregated data is not suited for predicting the response.

Overall, this study shows that machine learning can increase the predictive performance compared to GLMs. The results also indicate that aggregated data have the potential in terms of predicting claim occurrences, and can be used as a supplement in the actuarial world of risk assessment.

Acknowledgements

This thesis is written as part of our Master of Science in Economics and Business Administration, with respective majors in Financial Economics and Business Analytics, at the Norwegian School of Economics (NHH). The thesis constitutes 30 ECTS and conclude our master's degree.

Writing this thesis has been a rewarding experience and an exercise in patience, persistence and learning ability. We chose this topic based on both our individual interests in digitalization, technology, machine learning and data science, and the research has given us a great opportunity to further enhance our competence. Faced with challenges such as data processing, model implementation and interpretation, the development of this thesis has given us the outmost respect for the efforts, nuances and expertise required in these fields.

We want to express our sincere gratitude to our supervisor, Lars Jonas Andersson, for your guidance, feedback and support through the development of this thesis. Furthermore, we would like to thank Frende Forsikring for trusting and providing us with the data necessary. In particular, we would like to thank Anders Dræge and Eivind Herfindal Reikerås for sharing your intuitions, comments and insights for the past months. You have given us valuable knowledge into new aspects of the insurance industry. Last, but not least, we want to express our deepest gratitude to our friends, family and partners for your support not only through the development of this thesis, but throughout our academic journey. Without you, the path to where we are now would have been immeasurably more difficult.

Norwegian School of Economics

Bergen, May 2022

Contents

List of Figures

List of Tables

1 Introduction

"There were 5 exabytes of information created between the dawn of civilization through 2003, but that much information is now created every two days."
- Eric Schmidt, Executive Chairman at Google, 2010

Insurance as a concept can be dated back to ancient times in human history, where so-called bottomry contracts were granted by Babylons in ancient times. Bottomry contracts were loans granted with shipments as security. The loan was dropped if the shipment was lost during its voyage, and therefore the interest covered the insurance risk (Trenerry, 2009). Through the development of large economies and established financial institutions, the insurance industry established itself as one of the building blocks of modern society. Today, the insurance industry generates a global gross premium of $5.23 trillion a year (OECD, 2020a).

Insurance companies offer economic protection to its customers in exchange for a yearly fixed premium, often determined through differentiated pricing. It is therefore crucial for companies to assess aggregated risk and total expense related to the customer and the underlying individual policy. The insurance market is characterized by fierce competition, both internationally and in Norway, with companies perceived homogeneous by policyholders. It is therefore important for an insurer to offer fair premiums whilst still being able to cover its expenses. If one sets prices to high, one either risk losing customers to competing companies, or expose oneself to only obtaining customers with high risk, known as adverse selection.

The insurance industry is particularly data driven, and new big data technology can greatly influence the way insurance companies manage and analyze their data (Boodhun and Jayabalan, 2018). Generalized Linear Models has been the conventional method utilized in actuarial science, both for predicting claim probabilities and claim severity (size). According to McKinsey Global Institute, the insurance industry has lagged behind with regards to applying big data analytics compared to other industries in the financial sector (Clarke and Libarikian, 2014).

With a significant increase in computational power over the past decades, and a great potential in the usage of big data, there has been a great interest in applying machine learning algorithms to challenge the actuarial comfort zone of Generalized Linear Models. In this thesis we will therefore apply machine learning techniques to classify claim occurences and estimate claim size using aggregated data on corporate insurance customers, employing data provided by Frende Forsikring. Additionally, we will use the predictive models to possibly identify preferable or non-preferable customers with respect to both risk and potential profitability gains.

2 Background

2.1 Background and Motivation

Frende Forsikring is an insurance company located in the western part of Norway. The company consists of two subsidiaries, Frende Skadeforsikring AS and Frende Livsforsikring AS. Frende was founded in June 2007 and has since experienced a substantial growth, with a total of 250 000 customers in the private and corporate markets.

Today, Frende utilize Generalized Linear Models and have well-performing models, especially for private policyholders. Additionally, Frende has also experimented with advanced machine learning models for this customer group. The corporate insurance market is, on the other hand, a market with a greater potential for improvement in terms of predictive modelling and risk assessment. Thus, this thesis covers an analysis on Frende's corporate policyholders.

The risk assessment of a customer, and consequently setting the premium, is conducted on an individual policy level. For instance, the premium for a company car with collision damage waiver is set by applying risk assessment models developed specifically for that underlying policy type, with the relevant input information from the respective customer. Consequently, the total premium for a single customer is the sum of all individual premiums retrieved from the individual assessment models. The total premium is accordingly related to the total expected cost of a customer, thus this amount encaptures the probability of claim(s) and the severity if such claim(s) were to take place.

Through talks with Frende, we gained insights into the possibilities of using aggregated company characteristics and insurance data to assess the company's overall claim probability and total claim severity. We were motivated to explore whether such data, which we have termed aggregated data, could also be used to potentially identify good or dissatisfactory customers with respect to future profitability through predictive modelling using different machine learning methods. Aggregated data is, in this case, data that summarizes a company's characteristics, total premiums, number of claims, claim sizes and the policies they hold.

The first goal of this thesis is ultimately to explore whether company characteristics on an aggregated level have any additional predictive capabilities in terms of predictive modelling on corporate insurance customers, compared to only applying risk assessment on individual policy levels. Under the assumption that a corporate customer's individual policy risk is substantially represented through the premium, we will do this by building predictive machine learning models that models a customer's claim occurrence and total claim severity. This will allow us to assess whether aggregated data contributes with additional explanatory power, compared to predicting the response with premium as the only input variable. Modelling claim occurrences will be treated as a binary classification problem and denoted as *Claim* (α) throughout the thesis, whilst the claim serverity will be modelled as a continuous variable denoted *Claim size* (β). The terms *Claim* and claim occurrences will be used interchangebly throughout, and the same applies for *Claim size* and claim severity.

To explore the potential gains of using yearly company characteristic data and historic records of the customers, we apply every model on three sets of input variables: one containing only the premium, one containing the aggregated data, and one with both aggregated data and feature engineered time variables. The foremost will be denoted ***Yearly premium*** whilst the two latter will be referred to as ***Yearly variables*** and ***Yearly + time variables***, respectively.

The second goal of this study is to evaluate the additional gain of implementing machine learning algorithms compared to traditional method of Generalized Linear Models. To assess this CARTs, Random Forest, XGBoost and Neural Networks will be implemented in addition to GLM as a benchmark model.

Along with the research objectives, the study will touch upon the practical utilization of the selected best performing models through a experimental setup. This will be executed by combining the two models and assessing the resulting predictions. Combining the classification and regression models allows the exploration of whether company characteristics on an aggregated level have any additional predictive capabilities in terms of predicting total expected costs C. The combined model can be defined as the following:

$$C_i = \alpha_i * \beta_i \qquad where\, \alpha \in \{0,1\}\, and\, \beta > 0 \tag{2.1}$$

Essentially the total cost only encaptures the costs of corporate customers that are related to claims. In cases where there are no claims (α is equal to 0), there are by this definition no total cost related to the customer. Whilst, if one or several claims occurs (α is equal to 1), the total cost will be equal to the *Claim Size* β.

To illustrate practical usage of the predictive models, we will investigate the combined model's (Equation 2.1) ability to identify good customers. In order to do this, we must define what a good customer is in the context of Frende's operations. The perfect customer is one that has no claims. However, a customer with claims can still be viewed as profitable if the premium outweighs the claim size. Through talks with Frende, we got presented a rule of thumb saying that, through their domain knowledge, a customer with total costs (in terms of claims) less than 70% of the premium is deemed profitable. We therefore introduce the following model for identifying a good customer:

$$Premium * 70\% > \alpha_i * \beta_i \tag{2.2}$$

It is important to note that the implementation of the combined model is experimental. The combined model and corresponding analysis, is included to enhance the overall comprehension of machine learning techniques and aggregated data in relation to the two main research goals.

Whilst the motivation for the work and the data is related to Frende's operations, this thesis aims to contribute in assessing predictive modelling on corporate insurance customers for the industry as a whole. To summarize, the problem statement of this thesis is split in two as follows:

(1)*Does aggregated data containing company characteristic variables and time variables provide any additional information to the insurance premium when predicting claim occurrences and claim severity?*

(2)*Can machine learning increase the predictive modelling of corporate insurance customers compared to traditional GLMs?*

2.2 Related Work

Looking ahead, several insurance companies are exploring the possible value of different forms of data and machine learning for predicting claims and risk assessment in general (OECD, 2020b). This includes Frende, which is currently using models on aggregated customer data in their B2C market as a supplement to the individual risk assessment models. However, there is a substantial gap between what is currently explored internally among the insurance companies and published research material on the topic. Thus, the relevant available published work in this field is related mostly to the use of machine learning in predictive modelling on an individual policy level.

Blier-Wong et al. (2020) reviews current publications investigating the applications of machine learning models on ratemaking and reserving within property and casualty insurance (P&C). Their overview of related work presents 77 publications from 2015 to August 2020, with an increasing trend, especially after 2017. This comprehensive review, also touching upon, among others, studies from Frees et al. (2014), highlights significant individual differences among the insurance holders, and that machine learning models are useful in capturing this heterogeneity. Subsequently, they can be helpful for computing the premiums to reflect the true individual risk accurately. In addition to published work, the results of insurance pricing in competitions hosted on Kaggle have also been reviewed. This work presents XGBoost and Gradient Boosting Trees as the most popular pricing frameworks (Blier-Wong et al., 2020).

One of the earlier and larger studies related to the use of machine learning in predicting expected claim size is presented by Dugas et al. (2003). This study compares several statistical methods for ratemaking for automobile insurance. The work includes models in the families of linear regression, Generalized Linear Models, Decision Trees, Support Vector Machines and Neural Networks. The results showed promising results for Neural Networks, and the author encourage actuaries to include Neural Networks in their ratemaking models for car insurance. However, it took more than ten years after the work was published for Neural Networks to significantly experience an increase in popularity (Blier-Wong et al., 2020).

Guelman (2012) explored the usage of Gradient Boosted Trees compared to conventional GLM for loss cost insurance pricing modelling within the field of auto insurance. The research was conducted by solving both regression and classification problems with a high dimensional dataset. The gradient boosted predictions were higher than that of GLM and the author argued that gradient boosting might be preferable to other machine learning methods such as Neural Networks and Support Vector Machines due to the interpretability of the former and the black-box nature of the latter. Therefore, gradient boosting were presented as a good alternative to GLMs in terms of building loss cost models.

There has also been published work related to the probability of claim or claim frequency. Several studies treat claim occurrences as a binary classification problem. One of these studies is the research done by Bärtl and Krummaker (2020) on predicting claims for export credit insurance. Their research was performed on data from The Berne Union with the aim to accurately predict insurance claims. The study utilized four machine learning models being Random Forest, CARTs, Neural Networks and Probabilistic Neural Networks, and analyzed their performance among binary claim classification and claim ratio (defined as claim vs exposure). Random forest performed significantly better than the other prediction methods on all response variables. However, the authors expressed that the prediction for claim ratios were dissatisfactory.

Hanafy and Ming (2021) have performed a similar analysis as Bärtl and Krummaker (2020) for auto insurance. The research was performed on data provided by Porto Seguro, a large Brazilian automotive company, containing 59 variables and 1 488 028 observations. The observations included customer information and details about the respective insured car(s), over the years of the customer relationship. Predicting claim occurrences was treated as a binary classification problem, with the response variable holding the variable 1 if there had been a claim occurrence and 0 otherwise. The results show that Random Forest significantly outperformed logistic regression, Decision Tree, XGBoost, naïve Bayes and K-NN models with an AUC of 0.84.

In total, the list of literature related to predicting claim size and probability of claim with machine learning is small, but growing. It is also important to note that this review is by no means exhaustive, however, the most important topics related to this thesis are included. This thesis will contribute to this increasing list by utilizing complex tree-based algorithms and Neural Networks whilst comparing these methods to the generalized linear models when predicting claim occurrences and severities. Additionally, it contributes to the research of whether aggregated company data provides any additional information on the customer‘s risk profile, compared to setting the premiums through individual risk assessment. Thus, investigates if company characteristics adds explanatory power when predicting the risk associated with corporate customers.

3 Theory

Artificial intelligence (AI), machine learning and deep learning are terms that have gained considerable traction during the last couple of decades. Although often used interchangeably, there are key differences between them. AI is an umbrella term for systems or machines that imitate human intelligence. Machine learning is a subgroup of AI which provides systems and models that learn from data without being explicitly programmed (Chollet, 2021). Within the field of statistics, machine learning often falls under the definition of statistical learning through algorithmic modelling.

Machine learning can be divided into two main categories: supervised and unsupervised learning. Supervised learning train on labelled data where a set of inputs have influence over a set of outputs. In contrast, unsupervised learning trains models and learns from non-labelled data by utilizing for example clustering. This thesis falls within the aspects of supervised learning. With supervised learning, variables are often characterized as qualitative or quantitative. Predicting a qualitative and quantitative response is often referred to as classification and regression, respectively (Hastie et al., 2009) . We will use these terms throughout our thesis.

The methods utilized in this thesis vary in flexibility and interpretability. Flexibility can be seen as a model's capability to represent an output function f through a range of shapes (James et al., 2013). Linear regression through least squares is a model that has relatively low flexibility, but is highly interpretable in terms of understanding the relationship between the predictors. In contrast, deep learning with Neural Networks is a highly flexible method as it can generate a wider range of possible shapes for the output function f at the cost of interpretability with regards to the relationship between the predictors. In short, less flexible methods are often easier to interpret and are preferred if inference is the goal. Therefore, it is often a trade-off between flexibility and interpretability (James et al., 2013). Consequently, this thesis focuses on the predictive capabilities of the models such that we do not put emphasis on a model's interpretability.

The following subsections will briefly present the main ideas between the machine learning methods applied in this thesis. The models are presented in ascending order in terms of flexibility: Generalized Linear Models, Classification And Regression Trees (CART), Random Forest, XGBoost and Neural Networks. Lastly, this section will present the challenge of an data set imbalance and suggest methods to overcome this issue.

3.1 Generalized Linear Models

Generalized Linear Models (GLM) was first introduced by Nelder and Wedderburn (1972), and is an expansion of linear regression by utilizing flexible generalization. GLM often refers to a large class of models that allows the response variable to follow a probability distribution from the exponential family, usually referred to as the distribution family.

GLMs consist of three components: a random component, a systematic component and a link function (Dobson and Barnett, 2018). The random component connects the response variable with a distribution family. The systematic component is the linear combination of the explanatory variables. Lastly, the link function η connects the random and the systematic component by specifying the relationship of the expected value of the response (regarding the probability distribution) with the linear combination of predictors. To exemplify, simple linear regression falls within the GLM framework with the random component assumed to be normal distribution, a linear combination of the explanatory variables and an identity link function $\eta = E(Y)$. Estimation of a GLM is, in this thesis, done through maximum likelihood estimation using IRLS (iteratively reweighted least squares).

GLMs can be used with both binary and continuous response variables (Dobson and Barnett, 2018). Logistic regression is a subcategory in the GLM framework and is often used on classification problems, thus it will be used for modelling *claim* in this thesis. Logistic regression assumes a binomial probability distribution (random component) and a log-odds link function $\eta = log(\frac{\pi_i}{1-\pi_i})$, where the expected value of the response variable has a probability of π. The logistic regression can be written as:

$$log(\frac{\pi_i}{1-\pi_i}) = \beta_0 + \beta_i x_i \tag{3.1}$$

Furthermore, this thesis also utilizes GLMs on a continuous response variable. Due to uncertainty regarding the nature of the variable and with domain knowledge from Frende, this thesis will evaluate both inverse gaussian and gamma as distribution families.

Examples of the distributions are displayed in Figure 3.1, whilst the respective link functions are presented in Table 3.1 (note that two link functions are tested for each distribution family). In terms of feature selection there are several methods that can be applied, however, this study will conduct *L1* regularization with lasso regression. Lasso regression use a penalty term λ that shrinks the coefficient parameters towards zero, thus can be seen as a form of variable selection (James et al., 2013).

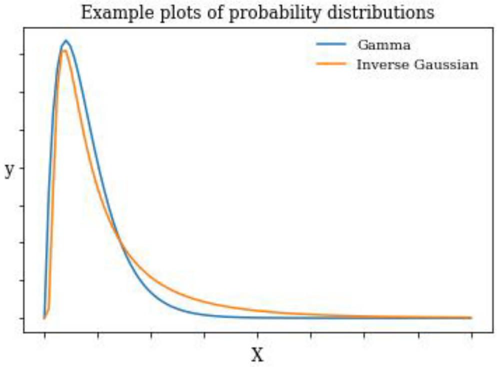


Figure 3.1: Distribution families

Distribution family	Link function
Gamma	$\frac{1}{\mu}$ and $\ln \mu$
Inverse Gaussian	$\frac{1}{\mu^2}$ and $\ln \mu$

Table 3.1: Link functions

Generalized linear models have several advantages. Firstly, it allows for different probability distribution compared to regular linear regression and allows for domain knowledge of the response to play part in the modelling process. Secondly, the choice of link function may differ from the random component, yielding more flexibility. Lastly, the methodology allows for nonlinearity in the explanatory variables and can improve modelling when the actual relationships between the response and predictors are nonlinear (Dobson and Barnett, 2018).

3.2 Tree-Based Methods

Tree-based algorithms are highly popular and have been shown to perform well with different supervised learning applications such as regression and classification. In their initial state they are highly interpretable and simple to implement. The simplest form of tree-based algorithms is classification and regression trees (CART), although they are often outperformed by more advanced methods (Alpaydin, 2020). Therefore, this section will also introduce two advanced tree-based ensembles: Random Forest and XGBoost.

3.2.1 Classification and Regression Trees (CART)

These algorithms revolve around segmenting the predictor space into sub-regions, resulting in a model that is similar to that of a tree. To build a tree the algorithms searches the predictor and split that is the most informative of the target variable and constructs the tree in a top-down perspective. The top of the tree is called the root and represents the entire set of data, the points or splits along the tree is known as internal nodes. The sub-regions created are known as terminal nodes (leaves) and indicates the response (prediction) for an observation that falls within the region, every observation that falls within a region receives the same prediction (see e.g., Hastie et al., 2009).

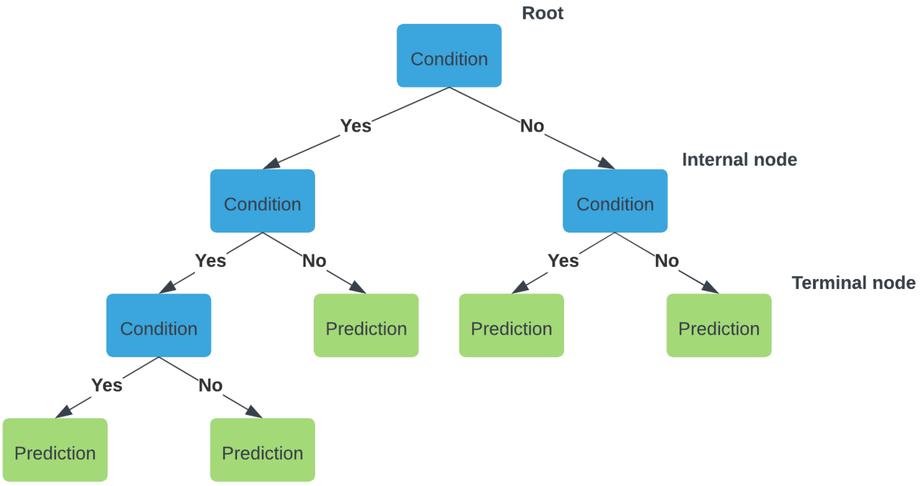


Figure 3.2: Example of a decision tree

When constructing for instance a regression tree, the goal is to segment the predictor space into $R_1, ..., R_J$ regions that minimizes the residual sum of squares:

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \tag{3.2}$$

In Equation 3.2, $\hat{y}_{R_j}$ refers to the mean of the training observations that lie within region j. Furthermore, the tree is built using binary recursive splitting, meaning that each split is decided by the one that has the greatest reduction to the residual sum of squares at that point. Consequently, future potential splits that give better performance overall are ignored. Classification trees are constructed in a similar manner, however, the predictor space is segmented by gini impurity rather than residual sum of squares (James et al., 2013).

The process of building decision trees often returns good predictions on training data, but often overfits the data. This is likely due to a too complex model, leading to a poor performance with regards to the test set (Hastie et al., 2009). Therefore, one often seeks a less complex tree to lower the variance and increase the bias. Cost complexity pruning provides an option for controlling the size of a decision tree. This technique initiates with a large tree and prunes the tree (creates a sub-tree) with a complexity tuning parameter α and a tree's complexity value $|T|$. The optimal α constructs a subtree that through cross validation returns the lowest prediction error (James et al., 2013). The process of building and pruning a regression tree can be described as following:

Algorithm 1 Building a regression tree with pruning (quoted from (James et al., 2013, p. 309))

```
1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations
2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of $\alpha$
3. Use K-fold cross validation to choose $\alpha$
    (a) Repeat Steph 1 and 2 on all but k$^{th}$ fold of the training data
    (b) Evaluate the mean squared prediction error on the data in the left-out k$^{th}$ fold, as a function $\alpha$
   Average the results for each value of $\alpha$, and pick $\alpha$ to minimize the average error
4. Return the subtree from Step 2 that corresponds to the chosen value of $\alpha$
```

One of the key downsides to using regular CARTs is that they, despite the usage of pruning, tend to overfit and provide poor generalized errors, as mentioned above. Therefore, ensemble methods are often used to provide better results (Alpaydin, 2020).

3.2.2 Random Forest

Ensemble methods are techniques that combine multiple machine learning models to produce a more powerful model. Random Forest, introduced by Breiman (2001), use a modification of a technique called bootstrap aggregation (bagging) to build a collection of decorrelated trees. Bagging is a technique that has the purpose to reduce the variance of a machine learning method by averaging multiple samples of the same model. This is done by bootstrapping i.e., selecting random samples of the training data with replacement. For instance, using bagging regression trees is done by selecting B bootstrapped training sets and training B trees and averaging the results (see e.g., Hastie et al., 2009).

Random Forest introduces a slight modification to the bagging procedure with the aim of decorrelating the trees constructed. This is done by selecting a random subsample of the total number of predictors m as candidates for each split of the tree. This method combats overfitting and contributes such that every variable has influence regardless of their initial perceived predictive power. To select the final prediction for an observation the majority vote of the ensemble is selected in the classification case, whilst the average is taken in the regression case (Hastie et al., 2009). The process of Random Forest can be described as follows:

Algorithm 2 Random Forest (quoted from (Hastie et al., 2009, p. 588))

```
1. for $b = 1$ to $B$ :
    (a) Draw a bootstrap sample $Z*$ of size $N$ from the training data
    (b) Grow a Random Forest tree $T_b$ to the bootstrapped data, by recursively
        repeating the following steps for each terminal node of the tree, until the
        minimum node size $n_{min}$ is reached
        i. Select $m$ variables at random from the $p$ variables
        ii. Pick the best variable/split-point among the $m$
        iii. Split the node into two daughter nodes
2. Output the ensemble of trees $\{T_b\}_1^B$
```

3.2.3 Gradient Boosting and XGBoost

Like Random Forest and bagging, boosting is an ensemble method with a different strategy. The main idea is to build new models in the ensemble sequentially. At each sequence, a weak learner is introduced to accommodate for the errors (pseudo-residuals) produced by the model up until that point (Natekin and Knoll, 2013). A weak learner is an algorithm that do not perform well by itself, but still performs significantly better than random guessing. CARTs are frequently used as the weak learner (Zhang et al., 2019). The resulting prediction from boosting is the sum of predictions made from weak learners in the ensemble.

Boosting algorithms vary in their design. Friedman (2001) introduced a gradient boosting machine as a further development of the established boosting algorithms at that time. Gradient boosting machines (GBM) utilize optimization of a differential loss function $L(y, f(x))$ to identify the pseudo-residuals r_{im} of the weak learners m by taking the derivative of the loss function with respect to a predicted value. For a set of pseudo-residuals, a new output value γ is calculated such that the loss function is minimized based on both previous predictions $f_{(m-1)}$ and the prediction for the residuals. The total prediction f_m is then updated with the additional weak learner. The concept of gradient boosting with regression tree as weak learner is showcased with the following algorithm provided by (Hastie et al., 2009):

Algorithm 3 Gradient boosted regression tree (partly quoted from (Hastie et al., 2009, p. 361))

```
1. Initialize with a constant value: $f_0(x) = arg\, min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$
2. For $m = 1$ to $M$:
    (a) for $i = 1, 2, ..., N$ compute the pseudo-residuals
            $$r_im = -\left[\frac{\delta L(y_i, f(x_i))}{\delta f(x_i}\right]$$
    (b) Fit a regression tree to the pseudo-residuals $r_im$ giving terminal regions $R_im, j = 1, 2, ..., J_m$
    (c) For $j = 1, 2, ..., J_m$ compute the output of the regression tree
            $$\gamma_{jm} = arg\, min_\gamma \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma)$$
    (d) Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$
3. Output $\hat{f}(x) = f_M(x)$
```

XGBoost (e**X**treme **G**radient **Boost**ing) is a further developed implementation of the gradient boosting algorithm and was first introduced by Cheng & Guestrin (2016). It has been one of the most popular machine learning methods in recent years, and has been among the best performing algorithms in multiple Kaggle competitions (Mello, 2020). XGBoost is an advanced machine learning algorithm with multiple features and nuances.

Consequently, the following will briefly present the main ideas of the algorithm.

XGBoost introduces CARTs as the weak learner, however, in contrast to the methodology presented in Section 3.2.1 it uses similarity and gain measures as split criterions in the tree construction process. Another key difference between Friedman's gradient boosting machine (Friedman, 2001) and XGBoost is that the latter utilizes both $L1$ and $L2$ regularization to avoid overfitting, thus improving the model's performance (Chen and Guestrin, 2016). XGBoost can be expressed mathematically by minimizing:

$$\begin{aligned} \mathcal{L}(\phi) &= \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \\ where \Omega(f_k) &= \gamma T + \frac{1}{2}\lambda \|\omega\|^2 \end{aligned} \tag{3.3}$$

In Equation 3.3, l represents a differentiable loss function similar to that of gradient boosting machines, and $\Omega(f_k)$ is the penalization term that regularize the complexity of the model. Each tree is represented by f_k where T is the number of leaves in a tree and ω represents the scores of the leaves. The regularization parameters λ and γ reduce the complexity of the model and represents $L1$ and $L2$ regularization, respectively (Chen and Guestrin, 2016).

One of the most important benefits of using XGBoost compared to gradient boosting is its scalability, as it can be less computationally demanding compared to previous gradient boosting techniques. Additionally, the system is designed to be able to handle sparsity in data, such as missing values and inflation of zero-values. Lastly, XGBoost introduce a vast amount of hyperparameters that can be tuned to further optimize model performance. (Chen and Guestrin, 2016)

3.3 Neural Networks

Deep learning has experienced vast progress in the last decade and has been applied to several different challenges such as image, text and voice recognition, as well as predictive modelling including classification, regression and time series analysis. Neural Networks are the foundation of deep learning, and its start can be dated back to 1943 (McCulloch and Pitts, 1943). Neural Networks are included in this thesis because they have shown the ability to handle complex and non-linear problems (James et al., 2013).

Neural Networks are network systems that are vaguely inspired by the field of neuroscience (Goodfellow et al., 2016). Today *feedforward* Neural Networks utilize an architecture that consist of one or several layers (Deep Neural Network). A Deep Neural Network can be broken down into an input layer with X_p predictors, hidden layers L_n and output layers $f_m(X)$. Within each layer there are K artificial neurons (or nodes), each of which has some learnable weights (w_n) connecting it to all the nodes in the previous layer. The inputs into a layer and the layers' corresponding weight matrix form a linear combination; To introduce non-linearity, a differential non-linear function, referred to as an activation function g, is applied to the linear combination. In a feed-forward manner the input is passed down the network, layer by layer, until finally the network forms a prediction in the output layer. Initially, the sigmoid activation function was common, however, in recent years the ReLU activation function is frequently used (Agarap, 2018). A method of fitting the parameters is the backpropagation algorithm, which uses partial derivatives and gradient descent to minimize a given loss function (see e.g., Hastie et al., 2009). An example of a Neural Network is shown in Figure 3.3:

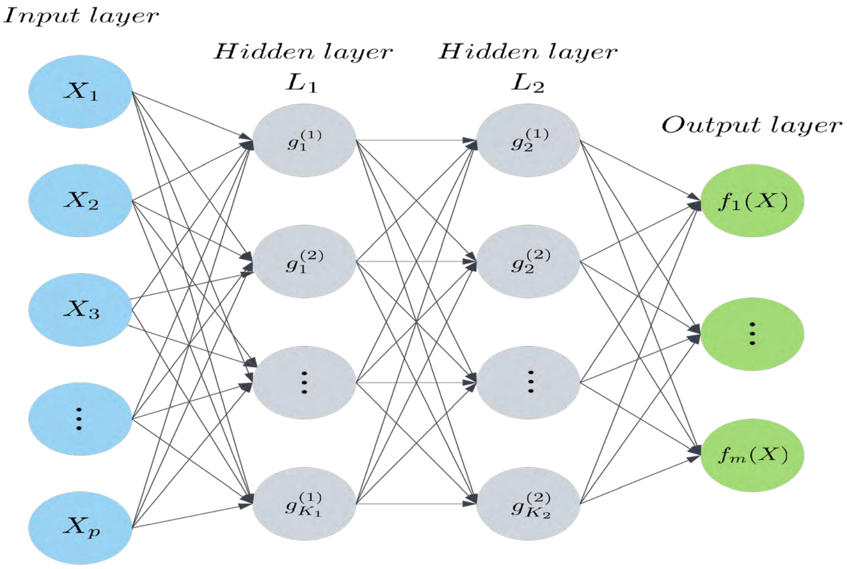


Figure 3.3: Neural Network illustrative example

Neural Networks have performed well in terms of pattern recognition, although they are often prone to overfit the training data, and therefore require substantial tuning. In terms of interpretability, Neural Networks can be seen as a black-box model, meaning that the model's complexity leads to low transparency of the predictors' relationship to the response (Goodfellow et al., 2016).

3.4 Imbalanced Data

Most of the statistical learning methods applied in this thesis assume that the underlying class is balanced with regards to the binary response variable (Krawczyk, 2016). In the context of binary classification imbalanced data refers to data in which one class is greatly overrepresented compared to the other. These classes are referred to as the majority and minority class, respectively (Fernández et al., 2018). In the presence of imbalance, machine learning methods can yield sub-optimal results, and frequently used performance measures might be deceiving (Chawla et al., 2004). The minority class is usually the one of interest. One of the reasons why a classifier returns sub-optimal results is due to possible overfitting to the majority class and the minority class being treated as noise in the data (Haixiang et al., 2017).

Furthermore, regular metrics used for validating a classification model's performance, such as accuracy, can be misleading in depicting a model's actual performance. This is known as the accuracy paradox. For instance, consider a situation where the majority class represents 95% of the observations, but the minority class is of considerable importance. If one were to implement a simple model predicting every observation as the majority class, it would return an accuracy of 95% which in some cases is deemed a good prediction score. In addition, with less information about the minority class, it becomes harder to retrieve substantially good predictions (Brownlee, 2020b).

There are several techniques to overcome the issue of class imbalance, such as collecting more data, resampling the data or utilize different performance metrics (Johnson and Khoshgoftaar, 2019). Resampling is one of the most common methods used to handle dataset imbalance and refers to either removing (undersampling) or adding (oversampling) samples to rebalance the dataset. At the simplest form this is done by either randomly removing samples of the majority class or randomly replicating samples of the minority class to the point where the proportion of each is equal. Although easy to implement, random oversampling might lead to overfitting of the replicated samples, whilst random undersampling might omit relevant observations from the majority class (He and Ma, 2013).

Due to these drawbacks, more advanced resampling techniques, such as *Synthetic Minority Resampling Technique* (SMOTE), are used to resample datasets (Brownlee, 2020b). SMOTE creates synthetic observations by interpolating n minority samples and randomly selecting a point between the samples to generate an additional observation. Furthermore, a resampling strategy of applying SMOTE in combination with a undersampling strategy has shown to produce better results than oversampling alone (Chawla et al., 2002). To illustrate, in a scenario with a 90% majority class, SMOTE can be used to partly oversample the minority class, whilst an undersampling technique completes the resampling by removing observations to a point where the data is balanced. This thesis will utilize SMOTE in combination with random undersampling as the resampling strategy.

4 Methodology

4.1 Data

The original data set from Frende contains 116 219 observations and 436 variables for each observation. There are 26 293 different companies represented in the data, with yearly observations ranging from 2008 to 2021. The variables can be categorized into explanatory and response variables. The explanatory variables are information available to Frende when calculating the insurance premiums for a given year, and the response variables are data that are not available until after the year has ended. All variables should be interpreted as variables for company i in year t.

Features included in machine learning models should be relevant and easy for the models to process (Zheng and Casari, 2018). Additional features have been generated through feature engineering to optimize the models' performance. This includes generating new variables as well as applying techniques to make the data compatible with machine learning algorithms. To further investigate the underlying data prior to pre-processing and modeling, Section 4.1.2 will present descriptive statistics of the most essential features.

4.1.1 Overview of Variables

A complete overview of the original variables is shown in Table 4.1 and 4.3. Additionally, Table 4.2 and 4.4 present the feature engineered variables. The complete overview of the data set after feature engineering and data pre-processing can be found in the appendix A1.

4.1.1.1 Response Variables

Original data set

The data set contains three response variables, as shown in Table 4.1. *Total cost* refers to the total amount paid out to company i in year t related to its claim, meaning that the variable represent the response variable of the combined model. *Number of claims* refers to the claims that are filed for, and *Approved claims* are the number of claims that Frende pays out. Henceforth, the latter is deemed the appropriate response used to determine the binary classification of *Claim*.

Response variable	**Description**
Total cost	Total amount paid out to the company, can consist of more than one claim, 0 when there are no claims
Number of claims	Number of filed claims
Approved claims	Number of claims accepted

Table 4.1: Response variables from the original data set

Engineered response variables

From the initial response and explanatory variables four additional response variables have been generated and are presented in Table 4.2. *Claim* is a binary value that holds the value 1 if approved claims is greater than 0, and 0 otherwise. Therefore, *Claim* forms the response variable for the binary classification models. *Claim size* represents the claim severity of the observations with claims, and is the response variable for the regression models. The potential claim severities of observations that have no claims are consequently unknown. The remaining three variables *Claim frequency*, *Claim probability* and *Claim percentage* describe the relative amount of recurrences and severity of the claim(s). These can be viewed as relevant historical explanatory variables that later will be feature engineered as time variables for a given observation.

Response variable	**Description**
Claim	1 if approved claims $>$ 0, 0 otherwise
Claim size	Equal to *Total cost* when *Claim* is 1, undefined otherwise (when there are no claims)
Claim frequency	Approved claims /Number of policies
Claim probability	For each company in year y: Number years with claim/Number of years as a customer
Claim percentage	Claim size / Insurance premium

Table 4.2: Engineered response variables

4.1.1.2 Explanatory Variables

Original data set

The data set contains 433 explanatory variables, whereas *Insurance premium* is expected to have significant explanatory value. Additionally, there are variables providing information on which policies a customers holds and company-specific variables, such as the number of employees and type of industry the business operates within. All variables are defined in Table 4.3, apart from all insurance policies. The data set has 412 variables representing the insurance policies. The variable values represent the number of policies from the specific product group an observation holds.

Explanatory variable	**Description**
Year	Year of observation
Insurance premium	Premium paid to Frende
Policy years	Sum policy years. Per policy;1 if the insurance is held in a whole year, 0.5 for 6 month etc.
Business type	E.g., joint-stock, foundation, county
Distribution channel	Where the customer was obtained
County	Location of company
Number of employees	Number of employees in the company
Noted customer	1 if abnormal claim size or claim frequency in the past, and/or missing payments, 0 otherwise
Foundation date	Foundation of company
Vat Registered	1 if Vat registered, 0 otherwise
Non-Profit Organizations	1 if registered, 0 otherwise
Credit score	Scored from 1-5 when the customer is obtained
Latest submitted annual accounting	Year of last submitted annual accounting
Business Address Land Code	Two letter country code
Bankrupt	1 if bankrupt, 0 otherwise
Under settlement	1 if under settlement (bankrupt), 0 otherwise
Insurances	412 types of insurance products, each as variable represents the amount of policies company i has of product x
Industrial Classification Category	Name of classification subcategory (NACE)
Has claim last three years prior to Frende	1 is claim prior to being a customer, 0 otherwise
Register of business enterprise	1 if registered, 0 otherwise
Foundation registered	1 if registered as foundation, 0 otherwise

Table 4.3: Explanatory variables from the original data set

Engineered explanatory variables

Additional explanatory variables have been generated through feature engineering, some as supplementary variables and some to replace the initial variables, as they would have been significantly correlated otherwise. All engineered variables are presented in Table 4.4.

Explanatory variable	**Description**
Main industrial classification category	A-U, NACE
Company age	Year - foundation date
Total number of policies	Sum insurance policies
Customer length	Number of years as a Frende-customer
Number of policies t-1	
Approved claims t-1	
Claim size t-1	
Claim frequency t-1	
Claim percentage t-1	
Number of policies t-2	
Approved claims t-2	
Claim size t-2	
Claim frequency t-2	
Claim percentage t-2	
Prior three years average claim frequency	
Prior three years average claim percentage	
Prior three years probability of claim	
Prior three years average number of policies	
Prior three years average number of claims	
Prior three years average claim size	Only included defined claim sizes when averaging
Defined number of employees	
Delta number of policies y-1	Number of policies today – last year
Delta number of policies y-2	Number of policies today – two years ago

Table 4.4: Feature engineered explanatory variables

Instead of having 572 industrial classification categories, these have been replaced with their main NACE code (ranging from A-U), to bring forth fewer instances. *Foundation date* has been replaced with *Company age*, as it is easier to interpret. In addition the variable *Total number of policies* has been created and indicates the total number of policies the customer i holds in year t. *Customer length* indicates the number of years each company has been a customer.

The insurance products are represented in a total of 412 variables. All insurance policies are represented in terms of subcategories of larger insurance categories. As there is a substancial number of insurance policies, the policies with less than 5% representation have been combined to collective subcategory within each main insurance category. To illustrate, take for example the main category of fire insurance. Within this category are policies such as *fire insurance - building*, *fire insurance - commercial building* and *fire insurance - agriculture*. Each of the two latter subcategories had less than 5% representation and have been combined into a *fire insurance other* subcategory. The two original variables have consequently been removed from the data set. Subsequently, the variables representing insurance policies are represented through 130 variables compared to the original 412. A complete overview of all included insurance products in the modelling of *Claim* and *Claim size* can be found in appendix A2.1.

To investigate whether historical data are of value in predicting the response variables, several time variables have been created using the responses from previous years. Variables ending with *t_ 1* and *t_ 2* in Table 4.4 are the response variables from last year and two years ago, respectively. In addition, both *Delta Number Of Policies variables* (*y_ 1* and *y_ 2*) describes the increase or decrease in insurance policies compared to one or two years ago. The last category of time variables is the average of the response variables of the prior three years. These are created due to the expectation that the last three years could depict a representative picture of an observation's trend if such a trend exists.

4.1.2 Descriptive Statistics

Descriptive statistics is useful to further investigate the data set prior to data cleaning and modeling. Descriptive statistics offers great insight into the available data, helps discover potential issues with the data, and is an important step in making sure the data is compatible with the intended models (Brownlee, 2016). First, we will introduce descriptive statistics for the numeric variables, before going further in depth with our most relevant features.

Table 4.5 shows descriptive statistics for numeric variables. Both *Insurance premium* and *Claim size* contain large amounts of variation. As expected this variation is notably larger for *Claim size* than for *Insurance premium*, as *Insurance premium* encaptures prediction for *Claim size*, thus *Claim Size* can have higher potential values. *Approved claims* >0, shows that if there exist at minimum one claim, the number of claims in the majority of the observations are 1 or 2. In addition, the maximum values for *Claim percentage* and *Claim frequency* are seemingly extreme, and should be addressed further. This could indicate that there exist some observations with extreme values for *Claim size*, *Approved claims* or *Insurance premium* that should potentially be removed. There also seem to exist some values in *Insurance premium* and for *Claim size* that are lower than expected, which will be addressed in the data cleaning section.

Variable	**Median**	**Mean**	**St. dev.**	**Min**	**25%**	**75%**	**Max**
Total cost	0	11777	140045	0	0	0	13452680
Claim size	19538	83882	365567	0.001	6502	57438	13452680
Approved claims	0	0.21	0.86	0	0	0	72
Approved claims (>0)	1	1.52	1.81	1	1	2	72
Claim percentage	0	0.56	8.03	0	0	0	694
Claim frequency	0	0.14	1.40	0	0	0	365
Insurance premium	9746	21312	73565	2.95	3759	22488	7304019
Policy years	1.63	2.61	3.57	0.002	0.97	3.00	121
Credit score	3	2.83	1.55	1	2	4	5
Company age	8	13	19	0	5	12	217
Customer length	2	2.80	2.73	0	1	4	13
Number of employees	0	3.57	16.82	0	0	3	1621

Table 4.5: Descriptive statistic of all numeric variables

There is also relevant information related to the explanatory variables. 50% of the customers have around 1-3 policy years. *Credit score* looks to be normally distributed. A large fraction of companies (>25%) are businesses that have been operating for at least 5 years, and there is a fairly large variation within this feature (*Company age*). In 75% of the data, the customer relationship (*Customer length*) has a duration larger than one year. Hence, these companies will have time variables of *t-1* for at least one of their observations. Most observations will also have values for *t-2*, and representative values for the average of the prior three years. The *Number of employees* is 0 in the majority of the observations, with the largest company having 1 621 employees. The corporate customers in the data set are mainly small and medium enterprises. Some observations are expected to have 0 employees, however, the amount is a lot larger than expected. Consequently, this will be further addressed in the data cleaning section.

4.1.2.1 Response Variables

The respective distributions of *Claim* and *Claim size* are displayed in Figure 4.1, and illustrates that only 14% of all observations have one or more claims. This results in an imbalanced data set. For that reason, this needs to be further addressed with regards to data partitioning, additionally, potential need for resampling techniques should be assessed.

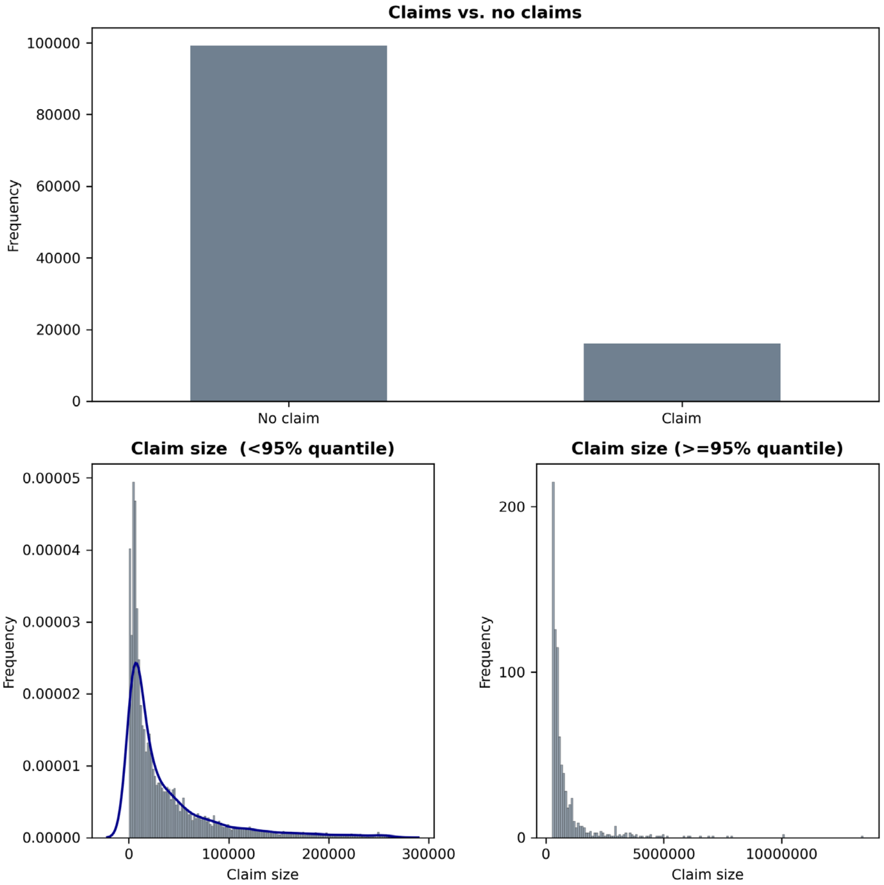


Figure 4.1: Response variables distribution

Figure 4.1 shows that *Claim size* has a steep and right skewed distribution. From the $< 95\%$ quantile, the probability is seemingly close to the gamma and inverse gaussian distribution refered to in Section 3.1. As shown in Table 4.5, the median for *Claim size* is 19538, whilst the mean is 83882, as a consequence of the skewed distribution. Furthermore, 95% of the observations are claims with an aggregated size less than 300 000, but the standard deviation is 365 567. The high standard deviation is a consequence of the variation in the largest 5-10% of the data.

4.1.2.2 Explanatory Variables

The data set contains a substantial amount of explanatory variables that have all been investigated thoroughly. In this section the assumed most important features are represented and discussed, including *Years*, *Insurance premium* and insurance products. These are important for modeling, in addition to further understand the context of predicting the response variables.

Years

Frende has experienced a linear and large growth from 2008 to 2021, more than doubling its customer base from 2012, as illustrated by Figure 4.2. Consequently, the data set will have fewer observations from the year 2008, and an increasing number of observations going towards 2021.

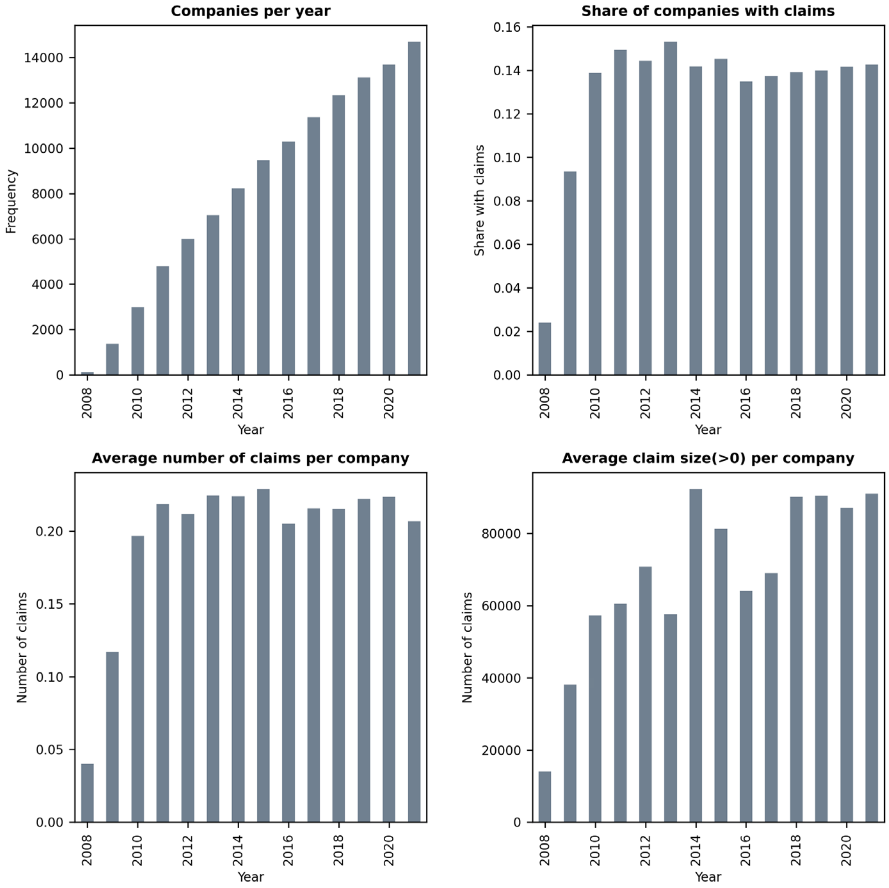


Figure 4.2: Development of response variables and number of companies from 2008-2021

There is a significant increase in both claims and the average number of claims after 2009. This might be explained by Frende taking on more riskier customers and/or offering products with a higher risk profile. There are some differences for these variables between the years after 2009, but seemingly without any trends for the share of claims, averaging at around 14%. The same is true for the average number of claims per company, which is around 0,20 after 2009. *Claim size* per company, given that there has been an approved claim, varies between the years, seemingly more than the number of claims and share of companies with claims. This could be random or related to different years having different severity in their claims, as a consequence of for example extreme weather. However, there seems to be a positive trend in the claim sizes, which could be due to for example economic factors, thus making *Year* a relevant explanatory variable.

Insurance Premium

The distribution of *Insurance premium*, a boxplot of *Claim percentage* and *Insurance premium* vs. *Claim size* is displayed in Figure 4.3. This visual analysis confirms the observations from Table 4.5. Most insurance premiums are below 74 000 (95% quantile), but there are also customers paying insurance premiums closer to 1 000 000 and above. In addition to what is displayed in Figure 4.3 of the insurance premiums larger than the 95% quantile, 27 observations have insurance premium ranging from 1 000 000 to 7 304 019, which have been removed from the plot to make it more readable.

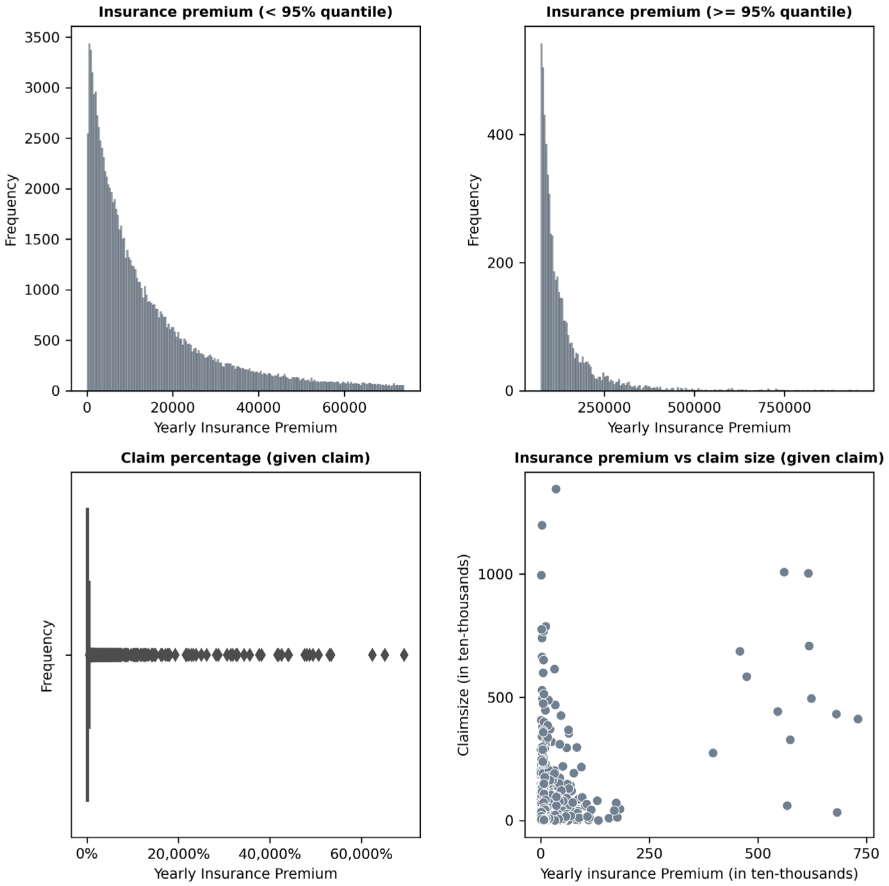


Figure 4.3: Insurance premium distribution and premium in relation to claim size

Given that there exists a claim, the median for *Claim percentage* is 67%. A customer is considered profitable if the claim size is less than 70% of the insurance premium, hence most companies with claims are in fact still profitable. However, *Claim percentage* has a mean 393%, and there exist cases with a claim percentage close to 70 000%, which is abnormally high.

From the scatter plot one can observe that there is not a perfect relationship between claim sizes and premiums, and it is unclear how correlated the variables are. The correlation value between *Insurance premium* and *Claim size* is 0.39. A correlation between 0.3 and 0.5 is often defined as low/modest positive correlation (Taylor, 1990).

Insurance products

To further investigate the various insurance products, these are displayed visually below. As the data is aggregated, it is difficult to untangle all details related to which claim is associated with which product. E.g., a data observation i can be a company having both product x and y and one claim z, and we cannot draw conclusions as to which of the products x or y that covers claim z. For descriptive statistics purposes all observations have been grouped in product types, meaning that all observations that have one or more policies of each of the 130 insurance products are grouped. Essentially the data observation i will be included both in the grouping of product x and for y with the claim z, even though in reality the claim is covered from x and with no connections to product y. The visualization of the products will give an indication of the most common insurance products and the products probability of claim. However, the implication of aggregated data is that this can also give a misleading picture of some of the product's types, such as for y in our example, thus conclusions from the figures should be drawn with care.

Figure 4.4 displays the most common insurance products among the observations in the data set. Subtypes within the categories business insurance, occupational injury insurance and vehicle insurances are the most frequent.

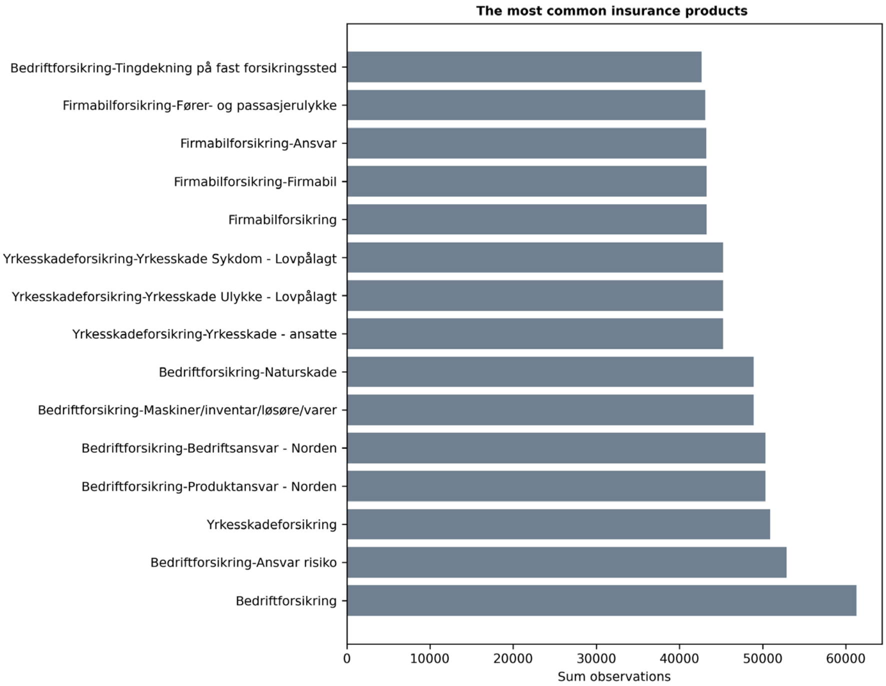


Figure 4.4: Top 15 most frequent insurance products

The insurance products with the largest number of related observations with (one or several) claims is shown in Figure 4.5. In the figure we see that several of the most common insurance products from Figure 4.4 are present. It makes sense that several of the most frequent products also have the most claims in absolute terms. Vehicular insurances are the products with the most related claims. In total close to 17 500 observations have car insurances for company cars and also have one or more claims. Approximately 45 000 observations are related to the most popular car insurance products (showed in Figure 4.5), meaning that the share of companies with claims of these insurance groups are getting close to 40%.

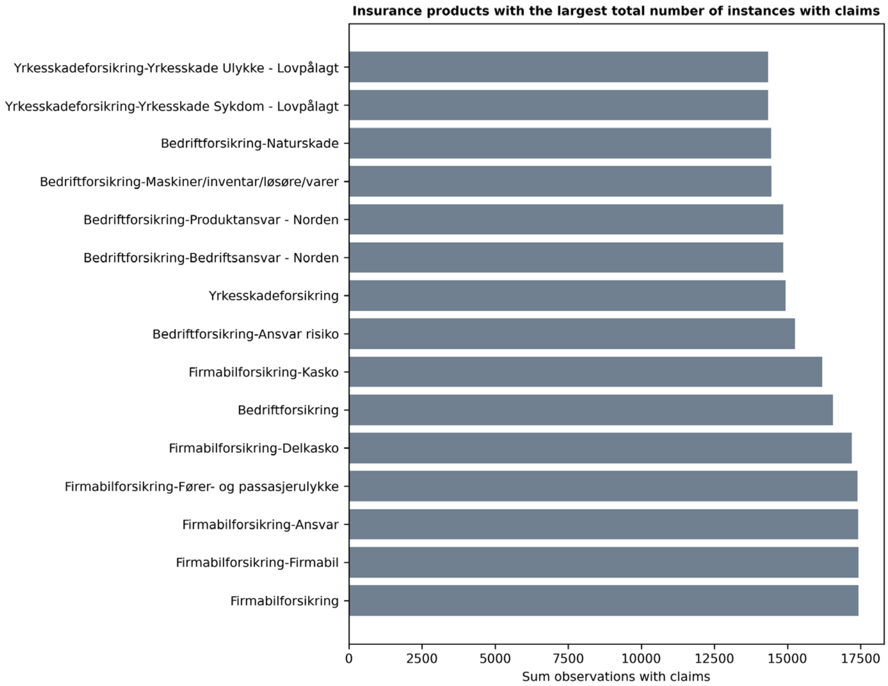


Figure 4.5: Top 15 insurance products with the most related claims

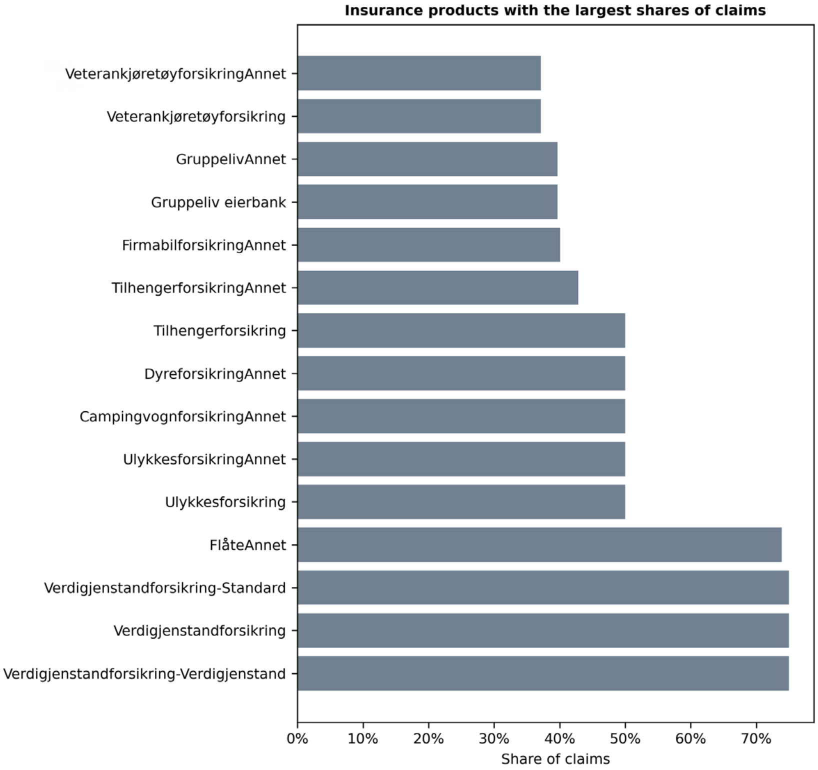


Figure 4.6: Top 15 insurance products in terms of largest shares of claims

Figure 4.6 illustrates the relative occurrences of claims for observations with the various insurance products in the data set. Less frequent insurance types are thus represented, with valuable articles insurance having the largest shares of claims. Insurance products related to vehicles are also present among the top 15.

4.1.3 Data Cleaning for Machine Learning

The data has been cleaned to enable compatibility with machine learning models and to eliminate errors in the data. In addition to simple data cleaning, such as removing duplicates, irrelevant variables and securing correct data types, we have focused on handling missing variables, outliers and categorical variables.

4.1.3.1 Missing Values

The data contains several variables with missing values. This study's predictive modeling techniques, apart from XGBoost, are incompatible with missing values and it is recommended to address this prior to the modelling (Brownlee, 2020a). One can handle missing data accordingly; remove the observation, remove the feature, impute the values, and/or create an indicative Boolean variable (Harrison, 2019). All methods have been applied in this thesis depending on the nature of the variable.

Removing observations
The simplest way of dealing with missing values is to remove the observation (Brownlee, 2020a). As several variables have a significant amount of missing values, removing all those observations would ultimately affect the amount of data severely. Hence, only the observations with missing values in the response variable *Total cost* were removed.

Removing variables
LatestSubmittedAnnualAccounts, *VatRegistered*, *RegisterOfBusinessEnterprise*, *Non-profit organizations*, *FoundationRegistered* are variables with more than 34% of missing values. Close to all observations that do not have missing values are registered in VAT and Business enterprise, and nearly none are registered as non-profit organizations or foundation. These are not expected to provide any essential information, especially as the variable *Business type* capture a lot of the same information. As a consequence of a high missing value rate, these variables are removed.

Imputing values

Imputing values is replacing the missing data with an estimated value, such that one can use the complete data as if the imputed values were actually observed values. Therefore, one must evaluate what one believe to be a suitable substitute value, as imputed values can introduce significant bias (Donders et al., 2006).

For the categorical variables *Credit score*, *Main industrial classification category* and *Business address landcode*, there are 0.3%, 1% and 26% missing values, respectively. These have been replaced by a new category__*missingvalues*__, which allows for the missing value information to be kept in the modelling process. Missing values within the binary features *Bankrupt* and *Under Settlement* have been replaced by the mode which in both cases are 0.

Company Age/Foundation date has 34% missing values, however, the feature is expected to have useful information. Therefore, the missing values have been predicted using K-nearest neighbors(KNN). This method is applied with the aim of reducing the bias compared to replacing the values with the mean or median. KNN has been proven to be an efficient imputation algorithm, replacing the missing data with the mean of the k nearest neighbors (Beretta and Santaniello, 2016). The KNN regressor is implemented with k equal to 5, using the Euclidean distance metric for calculating the distance between data points.

The feature engineered time variables have missing values when the observation has no historical record. To get a representative value for these instances, the variables have been replaced with the median of the relevant year. Meaning that e.g., for a company that became a customer in 2016, the variable *Claim size t-1* in 2016 was given the median of all defined *Claim size t-1* in 2016.

Indicative Boolean variable

For the *Number of employees*, 60% of the data points have the value 0. Although some observations are expected to have 0 employees, the amount is a lot larger than expected. This suggests that large parts of these observations are missing data. As this could be a relevant explanatory variable when predicting *Claim* and *Claim size*, and since the possibly missing value rate is this high, the problem was solved by including a new variable; *Defined number of employes.* The variable holds the value 1 if the *Number of employees* is larger than 0, and 0 otherwise.

Categorical Variables

The data set contains several categorical variables. These have to be transformed to numeric variables, as is the requirement for some machine learning methods (Zheng and Casari, 2018). The respective categorical variables have been one-hot encoded, meaning that all categories within a variable are represented as dummy variables. This has been performed on the variables *Business type*, *Credit Score*, *Distribution Channel*, *County* and *Business address landcode.*

Categorical variables can also be encoded by replacing categories with integers. This depends on whether the variable can be treated as a continuous variable or not (Müller and Guido, 2016). With the exception of *Year* and *Credit score*, the categorical variable's nature suggests that it is more proper to treat them as discrete values, hence, dummy variables. *Credit score* is a categorical variable, but as the score is from 1-5, with a higher score being the better score, one can argue to make this a numeric variable. However, under the suspicion that there might be a larger difference between a customer scored with 2 & 3 vs. 3 & 4, this variable has also been one-hot encoded. *Year* is on the other hand treated as a numeric value.

4.1.3.2 Outliers

The descriptive statistics in Section 4.1.2 showcased values in *Claim frequency*, *Claim percentage*, *Claim size* and *Insurance premium* that seemed abnormally high or low. The abnormal and extreme cases referred to in this thesis are cases that are considered to be so abnormal that there is a very high likelihood of errors in the data.

Then there is, however, the question of what should be considered an extreme value. A claim size larger than the 99,99% quantile is not necessarily extreme or abnormal, as it could be reflected in the company´s traits and risk assessment of the customer and insurance product, thus also reflected in a high insurance premium. *Claim Percentage*, representing the relative difference between *Insurance premium* and *Claim size*, is a better variable to evaluate. Moreover, *Claim frequency* is describing the relative amount of *Approved claims* compared to the *Policy years*, making it an interesting variable for the purpose of finding true outliers.

For skewed data, an interquartile range method is often recommended to identify outliers (Brownlee, 2020a). However, using this approach on *Claim Percentage* for observations with approved claims, identifies more than 12% of all observations with claims as outliers, which is suboptimal with regards to the framework of this thesis. Alternatively, one can also use anomaly detection for outlier detection, but the data set has too many features for this to be implemented in an efficient way. In other words, there is no precise way of identifying and defining outliers in every data set as the definition of an outlier is highly dependent on the underlying data. Therefore, one must interpret the data and decide which values are deemed outliers (Brownlee, 2020a).

Based on the findings in Section 4.1.2 of descriptive statistics, performing in depth analysis of the data and with domain knowledge from Frende, we have defined the following as extreme cases with high likelihood of errors:

1. Claim percentage > 20 000%
2. Claim frequency > 100
3. Very small insurance premiums (<100 (NOK))
4. Very small claims sizes (<100 (NOK))

These identified abnormal cases have been further investigated to confirm that it is reasonable to remove the observations. Based on the definitions, 37 observations were identified as abnormally large in terms of *Claim percentage* and *Claim frequency*, and 618 observations were identified as having a too small *Insurance premium* or *Claim size.* In total, removing these instances results in 655 observations being dropped out of the total 116 219 observations. Hence, there is a penalty when eliminating possible outliers, as the data set is already imbalanced and the imbalance is increased further by losing more observations of the minority class. This trade-off is important to consider, especially as tree-based machine learning models are in fact robust to outliers (Hastie et al., 2009). This is, however, also why the removal of outliers is as restricted as it is. Neural networks are somewhat more sensitive to outliers, which will be addressed in the modelling by standardizing the features (Goodfellow et al., 2016).

4.2 Evaluation of Models

There are several metrics that can be used to quantify a model's predictive performance. Unfortunately, there is no optimal metric for every situation and each metric has their individual strengths and weaknesses. However, depending on the problem at hand, some evaluation metrics generally performs better than others. Quantification of a model's performance is different in the context of classification and regression, such that our two response variables have different metrics applied to evaluate our model's performance.

4.2.1 Classification Metrics

In order to assess a binary classification model's performance there are several metrics that can be used. Among these are, for instance, metrics derived from a confusion matrix and the *Receiver Operating Area Under the Curve*, shortened ROC-AUC (see e.g., Hossin and Sulaiman, 2015). A confusion matrix is a visual representation of the predictive performance of a classification model. For a two class model, the confusion matrix consists of a two by two table that divides predictions into four categories: true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). True positives refers to the number of observations that are correctly classified as the positive class, whilst false negative are the number of observations that are incorrectly classified as the negative class. Analogically, the same applies for true negatives and false positives. The following figure describes a two-by-two confusion matrix:

	Actual Negative	**Actual Positive**
Predicted Negative	TN	FN
Predicted Positive	FP	TP

Table 4.6: Illustration of a confusion matrix

The objective of a classification model is to maximize the fraction of true negatives and true positives, thereby minimizing misclassifications. Accuracy is the most interpretable metric derived from the confusion matrix and details the percentage of the observations that are correctly classified:

$$Accuracy = \frac{TN + TP}{TN + FN + TP + FN} \tag{4.1}$$

The main advantage of accuracy is its interpretability. However, accuracy falls short when dealing with an imbalanced distribution of the predicted classes. This can be showcased using an example of predicting heart disease, where only 10% of the observations have the disease. By simply predicting every observation to not have heart disease would yield a 90% accuracy. A model with an accuracy of 90% would, with a balanced data set, be deemed a good model, but in this example the metric does not depict the model's true predictive capability. Precision and recall are two metrics that combat the disadvantage of solely using accuracy as a performance measure. Precision refers to the proportion of actual positive classification and is calculated as the ratio of correctly classified positives by the total of positive classified observations. In contrast, recall (sensitivity) refers the proportion of actual positive classes that are classified in the positive class (Hossin and Sulaiman, 2015).

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

Maximizing precision and recall will minimize the number of false positive errors and false negative errors, respectively. Therefore, precision may be an appropriate metric where false positives are of importance, whilst recall may be appropriate when false negatives are of importance (Brownlee, 2020b). It is important to note, however, that even though one error has importance, the other one should not be disregarded. To take both into concern one can utilize F-beta measure:

$$F_\beta = \frac{(1 - \beta^2 * Precision * Recall)}{\beta^2 * Precision * Recall} \tag{4.4}$$

A β-value of 1 refers to the F1-measure and treats the balance between precision and recall as equally important. If one sets a β-value of 0.5 precision is of more importance, whilst the opposite is true for a β-value of 2.

The receiver operating characteristics (ROC) is a probability curve and performance measure that is often used in machine learning. The ROC-curve plots the true positive rate versus the false positive rate at various probability thresholds for a given model, and summarizes the performance with respect to the positive class. The true positive rate (TPR) is equal to that of recall above, but is often termed as sensitivity in the context of ROC. The false positive rate is equal to $1 - specificity$ and is the proportion of observations that were misclassified as positive. The false positive rate (FPR) can be written as:

$$FPR = 1 - specificity = 1 - \frac{TN}{TN + FP} \tag{4.5}$$

Each point along the ROC-curve represents the false positive rate and the true positive rate at a given threshold between 0 and 1. Intuitively, each point also represents a distinct confusion matrix and corresponding measures derived from it. Examples of a ROC-curves is displayed in Figure 4.7. A ROC curve A is said to be dominative of another if it is above and to the left of curve B. However, there is often not a clear distinction between curves as they often perform differently compared to each other along the curve (Huang and Ling, 2005). This is illustrated in Figure 4.7 where ROC-curve A dominates B initially, but is outperformed with increasing FPR:

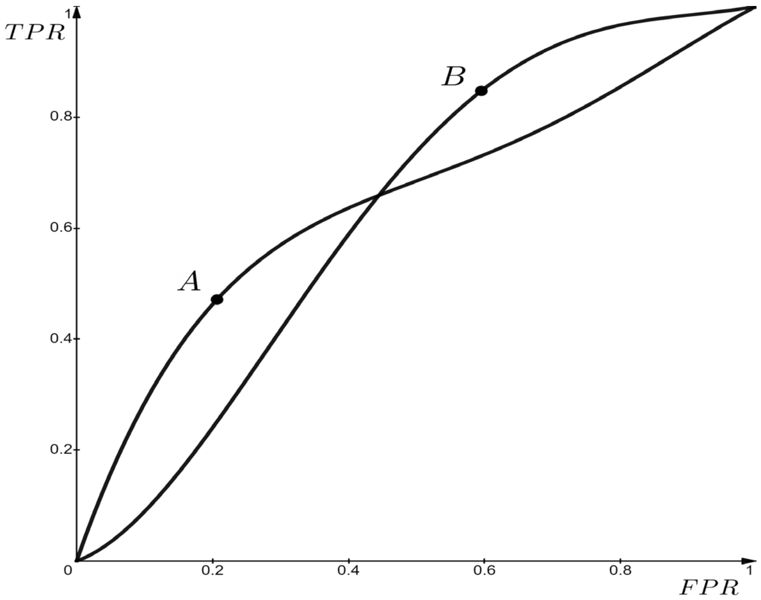


Figure 4.7: Example of ROC-curves where each performs best at different stages

To better distinguish between models, one of the most utilized measures is the area under the curve (AUC), which is calculated as the area under a given ROC-curve. The ROC AUC represents a model's ability to distinguish between classes for all probability thresholds (James et al., 2013). The ROC AUC has values between 0 and 1, where values close to 1 indicate a model with perfect ability to distinguish between the two classes. The higher the score, the better a model predicts the correct classes. Compared to a ROC-curve, the AUC summarizes a classifiers performance overall, and is therefore a great measure to compare different classifiers across all threshold values (Hossin and Sulaiman, 2015).

4.2.2 Regression Metrics

In contrast to classification problems, one cannot use the metrics presented above to evaluate regression problems on continuous variables. However, regression metrics are mostly easier to implement, interpret and understand. The most common evaluation metrics for numeric responses are mean squared error (MSE), root mean squared error (RMSE) and mean absolute error (MAE) (James et al., 2013).

Mean squared error (MSE) represents the average squared difference between the predicted and actual values (see e.g., James et al., 2013). MSE measures the goodness of fit, and the higher the error a model returns, the higher the MSE. It is is calculated by averaging the squared difference between observed value y_i and the predicted value $\hat{y}_i$. Root mean squared error represents the square root of MSE. RMSE is used more frequently compared to MSE due to interpretability. In some cases, MSE can become quite large number making relatability difficult. RMSE combats this by taking the square root, bringing the values back down to their original level. Additionally, large deviations are penalized more than smaller deviations and is therefore especially important if one does not want to emphasize such deviations. RMSE can be formulated as:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{4.6}$$

Mean absolute error is an alternative performance measure to the MSE and RMSE, and is a more direct representation of the errors a model produce. MAE represents the absolute deviation between the observed value and the predicted value. In contrast to RMSE, MAE does not penalize larger deviation, but rather treat all deviations equally. Using the same notation as above, MAE can be formulated as:

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{4.7}$$

RMSE is often preferred compared to MAE due to the usage of absolute value in the latter. Many machine learning models optimize its parameters through minimizing the differentiated loss function which is easily done by using MSE or RMSE. Optimization of MAE requires usage of different methods such as gradient descent which can be computationally demanding (Chai and Draxler, 2014).

4.3 Implementation

This section outlines the implementation of the presented models. This includes splitting of data, hyperparameter tuning, sampling techniques, as well as the software utilized. The process of writing this thesis has accumulated thousands of lines of code which in turn have been reviewed and corrected extensively to ensure the validity of our results. Lastly, the implementation is inspired by both academic literature and suggestions or solutions proposed on various forums such as Stack Overflow and GitHub.

4.3.1 Software

All technical aspects of this thesis, such as building models and processing data, have been applied mainly using Python. In addition, programming packages such as *scikit-learn* (Pedregosa et al., 2011), *Pytorch* (Paszke et al., 2019) and *XGBoost* (Chen and Guestrin, 2016) have been applied to implement the models. Specifically, scikit-learn has been used for pre-processing, cross validation, resampling, GLM (classification), CARTs and Random Forest. The Pytorch and XGBoost packages have been used for Neural Networks and XGBoost, respectively. All models have been implemented using Python. However, due to lack of flexibility and implementation difficulties, the GLM with *Claim size* as response had to be implemented in R with the *glmnet* library (Friedman et al., 2010). In this case, we ensured that the same training, validation and test sets were used by exporting them from our python scripts rather than doing the split in R itself.

4.3.2 Partitioning of Data

Evaluation of machine learning methods require both data to train and fit the model, and data to test the model's predictive capabilities on unseen data. If one were to both train and test models on the same data, the resulting evaluation scores would not depict that of reality. This means that the model is most likely overfit to the training data and consequently will not showcase its real predictive capability on new and independent data. Therefore, data are often split into a training set, a validation set and a test set. The training set is used to fit the model with optimized parameters that minimizes a selected loss function. The validation set is then used to evaluate the performance of the model. The test set is used to find the general error of the best performing model. In terms of model selection (comparing performance of different models), the validation set is used rather than the test set, such that the final chosen model does not underestimate the generalized test error (Hastie et al., 2009).

Typically, one would randomly select the data-splits by fractions of the total observations such that a 40-40-20 split refers to 40% used for training set, 40% used for validation set and 20% used as test set. There are several methods for choosing an appropriate split (see e.g., James et al., 2013). In this thesis we wanted to utilize a split such that we could implement our experimental model to analyze the potential gain from the models for a given year. Therefore, the last year (2021) of our data set is taken out in its entirety as our test set for the experimental model whilst a validation set is used to examine the predictive capabilities of our individual models.

For predicting *Claim* (classification), the remaining observations are split into a training set and a validation set by a fraction of 80% and 20% respectively. As a result, 65.46% of the total observations comprise the training set, 21.82% the validation set and 12.71% the test set. The same pseudo random number generator has been used such that the splits are equal for every model implemented across all scripts. Additionally, the split is stratified such that the proportion of claims in each data set remain equal. For predicting *Claim size* (regression), all observations that do not include response for *Claim size* (a value of 0) are removed such that the models train on relevant responses. Meaning that, as shown in Section 4.1.2.1, only 14% of the observations in the complete data set are included in the modelling process of *Claim size*.

The training data is used for fitting and hyperparameter tuning through cross validation. This process is further explained in the following section. The validation set is used for model selection for models predicting *Claim* and *Claim size*. The two selected models (from the validation set), are then combined and utilized on the test set. To be precise, the test set (data for 2021) is only utilized in the experimental setup to get an less biased estimate of the combined model's performance.

4.3.3 Hyperparameter Tuning

A vast majority of machine learning methods require some form of hyperparameter tuning in order to optimize the fit of a model and its corresponding predictive capability (Probst et al., 2019). Hyperparameters are parameters that are set manually by the user, meaning that they are not estimated by a model, but rather set prior to the model being fit. Machine learning methods have different set of hyperparameters, and the number of parameters varies. In short, hyperparameters restricts how the model learns and fits the data. A challenge with hyperparameters is that there is no initial "go-to" solution for a given problem, such that an optimal solution must be derived. Examples of hyperparameters are the number of trees in a random forest, the learning rate in XGBoost or the number of hidden layers in a neural network.

There are several methods to tune hyperparameters, but in this thesis we have utilized a grid search with 10^{th}-fold cross validation. A grid search, in this case, takes a predetermined subset of the parameters and iterates through every combination of the parameters in a 10^{th}-fold cross validation, and returns the average of a specified evaluation metric from each fold. Tuning hyperparameters using the grid search method is computationally challenging, especially with smaller increments in the subset of parameters. As a result, we cannot guarantee that the resulting parameters are optimal across all possible combination of parameters. However, they can still be assumed to produce results close to that of the optimal solution.

4.3.4 Model Development

This section will provide a brief overview of model implementation. The hyperparameters that returned the highest AUC for classification and RMSE for regression were selected for every respective model. A full description of the tuning process and consequent grid searches conducted are shown in the appendix A3.

In our models for GLM, we have used the l1-lasso regularization in order to reduce the number of variables in the fitted model. For classification, the logistic regression was used. When applying GLM on *Claim size* both the gamma and inverse gaussian were potential distribution families. Without the proper domain knowledge regarding the proper distribution family, both were tested and evaluated using RMSE. In contrast to other advanced machine learning algorithms, GLMs require little tuning efforts, however, we determined the penalty parameter λ through cross validation.

CARTs usually presents several hyperparameters to be tuned. However, they have been implemented through cost complexity pruning, resulting in a single hyperparameter α (pruning parameter). As CARTs are less computationally demanding compared to advanced machine learning techniques, the optimal α was found through several hundred candidates.

The Random Forest and XGBoost algorithms have multiple hyperparameters that can greatly affect the performance of the models. Therefore, the hyperparameters have been tuned extensively across a vast interval for each parameter. The following parameters were tuned for the Random Forest (note that they are denoted in the practical sense, not the theoretical): *Maximum depth, number of features selected, number of trees in the ensemble, minimum number of samples at each split and the minimum number of samples in each resulting leaf.* For XGBoost, we tuned: *Learning rate, maximum depth of a tree, the minimum sum of weights of all observations required in a child, gamma* (the minimum loss reduction required to make a split), *subsamples, fraction of columns to be randomly sampled for each tree, lambda* (L2 regularization term on weights) and *alpha* (L1 regularization term on weight).

The Neural Networks offers great flexibility and complex in terms of tuning. The models in this thesis have been tuned on *learning rate*, *number of layers* and *number of neurons*(in each layer).

We have developed all classification models both with and without resampling. The resampling has been conducted by using a combination of oversampling and undersampling, employing SMOTE and random undersampling, respectively. During hyperparameter tuning, resampling has been applied within the cross validation process, meaning that for each fold, the training data was resampled whilst the out-of-sample fold was kept as is. Lastly, the entire training data was resampled to fit the parameters of each specific model with optimized hyperparameters. It is important to note that our resampling strategy could also be treated as hyperparameters. However, due to already computationally demanding techniques, we have kept the strategy constant. Therefore, the training data was first oversampled making the proportion of the minority class (claim occurences) 35% of the total observations. Subsequently, the majority class was then randomly undersampled, matching the fraction of the majority class equal to that of the minority class i.e., a balanced data set.

5 Analysis of Results

This section will present the results for the classification models predicting *Claim* and the regression models predicting *Claim size*. Both classification and regression models will be presented and evaluated with the three different input-groups defined in Section 2.1:

Yearly premium – *Insurance premium* is the only explanatory variable

Yearly variables – *Yearly premium* + all variables related to the company and its products

Yearly + time variables – *Yearly variables* + feature engineered time variables

A complete overview of which of the variables presented in section 4.1.1 that are included in each of the three input-groups can be found in appendix A1.

The differences among the prediction results using the various input-groups will indicate if there is valuable information in company-specific characteristics, insurance products and time variables. This is in addition to information related to both claim severities and claim occurrences that is assumed represented indirectly in the *Insurance premium.*

5.1 Claim Occurrences

Predicting *Claim* is a binary classification problem, and is estimated with and without resampling techniques. The ROC AUC scores will be presented which will be the foundation for the model evaluation. As the outputs of the predictions will be binary, a threshold must be selected for classifying the observation into 1 and 0. Recall and precision metrics on the validation set will be presented after applying the optimal threshold in terms of F1-score and a less conservative threshold maximizing the F0.5-score. Both thresholds were selected in the training process to provide less biased results. Additionally, the variable importance for the preferred machine learning models for ***Yearly variables*** and ***Yearly + time variables*** will be presented as it will provide information on which features are the most important when predicting claim occurrences.

5.1.1 AUC Comparison

Table 5.1 shows the results in terms of AUC when predicting *Claim* without resampling techniques. The results of applying SMOTE and undersampling are shown in Table 5.2. In total these tables show that there are very modest differences among the two groups of models. Almost all algorithms and input-groups have marginally better performances when resampling is not applied. Nevertheless, in regards to AUC, there is no gain for the models to introduce resampling.

Model	Yearly Premium	Yearly variables	Yearly + Time variables
GLM binomial	0.7804	0.8198	0.8257
Classification Tree	0.7493	0.8011	0.8023
Random Forest	0.7390	0.8411	0.8425
XGBoost	0.7792	0.8413	0.8457
Neural Network	0.7803	0.8319	0.8335

Table 5.1: AUC results comparison without resampling techniques

Model	Yearly Premium	Yearly variables	Yearly + Time variables
GLM binomial	0.7804	0.8201	0.8255
Classification Tree	0.7502	0.7991	0.8019
Random Forest	0.7319	0.8402	0.8413
XGBoost	0.7789	0.8406	0.8434
Neural Network	0.7804	0.8300	0.8326

Table 5.2: AUC results comparison with resampling techniques

Predicting *Claim* solely on the basis of ***Yearly premium*** yields relatively acceptable results, with the AUC being above 0.73 for all algorithms. Including ***Yearly variables*** results in a substantial increase in performance for all algorithms. Meaning that when predicting *Claim*, there seems to exist explanatory power in the individual company-specific characteristics and/or insurance products. However, more surprisingly, there is only a small performance improvement when including the time variables for all algorithms. GLM(without resampling) has the largest improvement of 0.0059, and for comparison going from only ***Yearly Premium*** to including yearly variables yields an improvement of 0.0394. XGBoost without resampling, being the best performing machine learning model, only achieves an improvement of 0.0044 in terms of AUC when including time variables. This indicates that the likelihood of *Claim* is not that affected by prior years, i.e., there is likely no strong company-specific trends. It could also be related to a weakness in the imputation of the time variables when values are missing, affecting the performance of the features. Both ***Yearly variables*** and ***Yearly + time variables*** includes the variable *Has Claim Last Three Years Prior To Frende*, indicating that there could be some company-specific trend being captured in this variable as well.

The difference in the performance of the various machine learning models, visualized in Figure 5.1, is as expected when including all variables. Classification tree is the worst performing of the models, aligning with the empiric referred to in Section 3.2. The other models outperform the GLM model in terms of AUC when including additional variables to ***Yearly premium***, possibly indicating that the data has complex relationships that are easier for the more complex machine learning models to capture. Random Forest and XGBoost are very similar in their performance, with a slight edge to XGBoost. Neural network has a lower performance than the two complex tree-based ensembles.

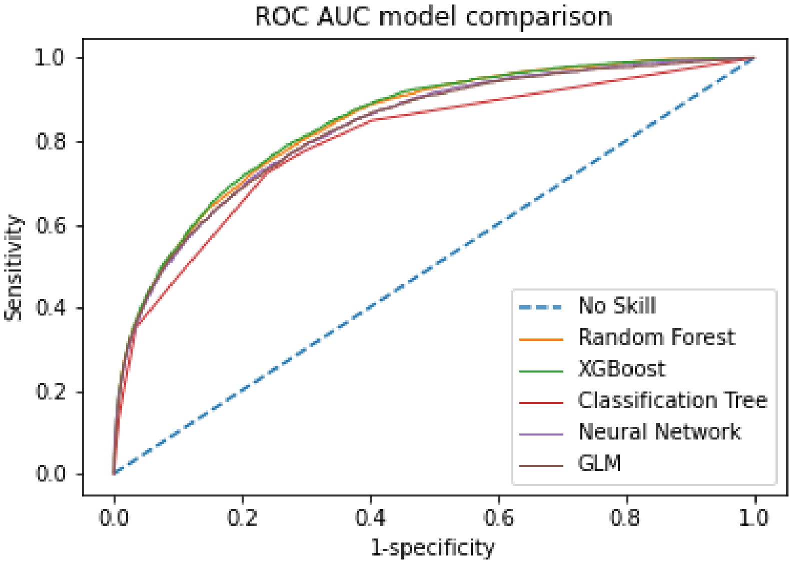


Figure 5.1: ROC-curve comparison modelling *Claim*

In total we see that XGBoost has the best performance and is marginally better when including time variables as opposed to only ***Yearly variables***. With an AUC of 0.8457 without using resampling techniques, we can conclude that the model is doing a reasonably good job at discriminating the two distinct classes.

5.1.2 Confusion Matrix & Metrics

To further investigate the effect of resampling and how well XGBoost classifies *Claim*, the confusion matrices for the models with and without resampling are presented with their corresponding metrics.

The thresholds maximizing the F1-score on the training set predictions are 0.332 with and 0.292 without resampling. The results applying these thresholds on the validation set are displayed in Table 5.3 and Figure 5.2 (a and c). If the value of precision is larger than recall, F-score with the beta 0.5 is more suitable to optimize. The latter gives the resampled model an optimal threshold of 0.592, whilst without resampling the data yields an optimal threshold of 0.443. Table 5.4 and Figure 5.2 (b and d) show the validation results for these, less conservative, thresholds.

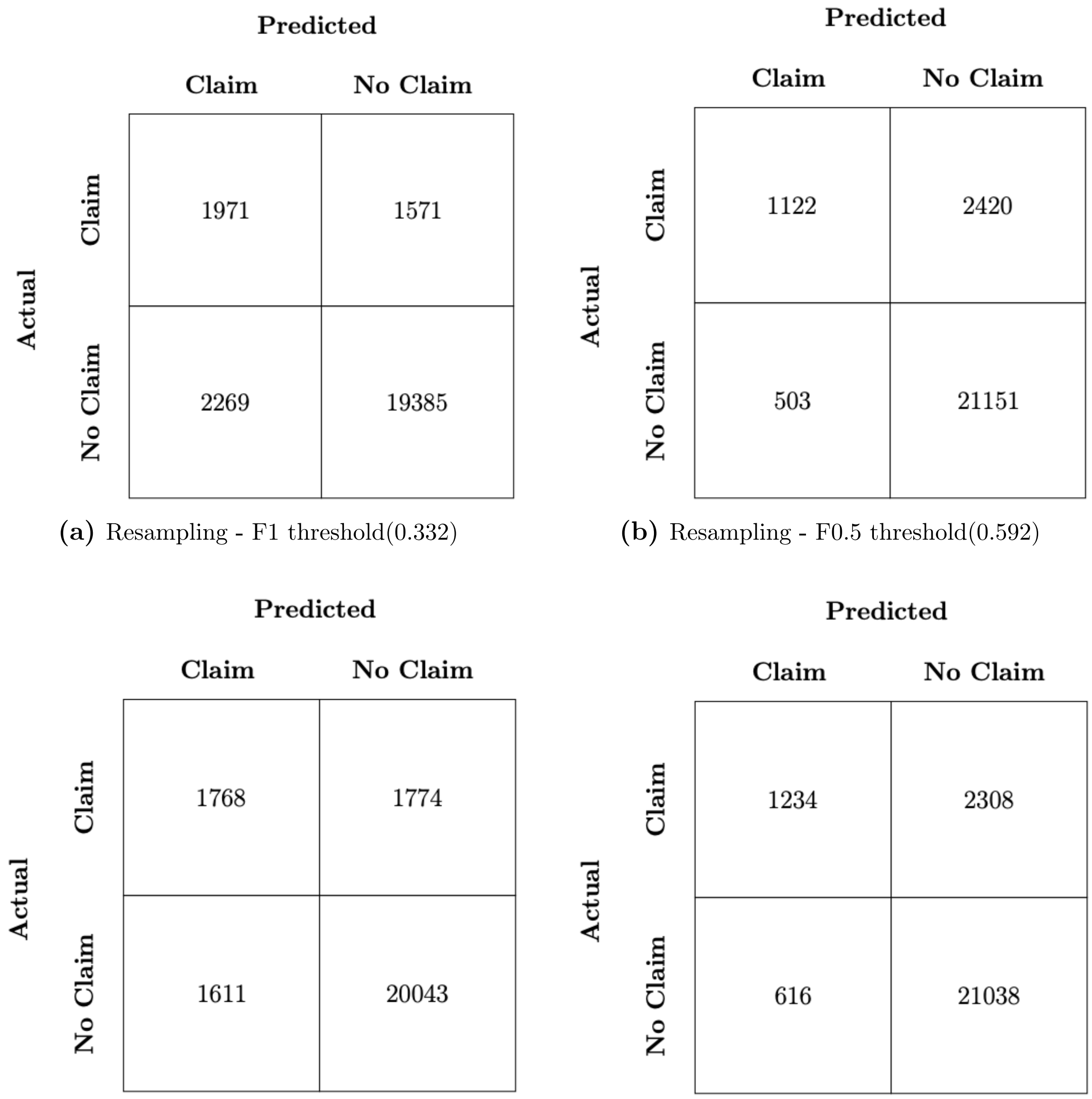
	Recall	Precision	Specificity	Negative predictive value	F1-score	F0.5-score
Resampling	0.5565	0.4649	0.8952	0.9250	0.5066	0.4807
No resampling	0.4992	0.5232	0.9256	0.9187	0.510	0.5182

Table 5.3: Confusion matrix metrics applying thresholds maximizing the F1 score (0.332 and 0.292 with and without resampling, respectively)

	Recall	Precision	Specificity	Negative predictive value	F1-score	F0.5-score
Resampling	0.3168	0.6905	0.9768	0.8973	0.4343	0.5586
No resampling	0.3484	0.6670	0.9716	0.9011	0.4577	0.5639

Table 5.4: Confusion matrix metrics applying thresholds maximizing the F0.5 score (0.592 and 0.443 with and without resampling, respectively)

The resampled model, with its optimal threshold in terms of F1-score, identifies 55.65% of the observations that will have one or several claims. However, the downside is that less than half (46.49%) of the predicted positives were actually correctly labeled. Alternatively, a more conservative threshold, such as maximizing the F0.5-score, can be applied. This results in a recall rate of 31.68% and precision rate of 69.05%. The precision increases, but the model identifies fewer of the data observations with claims.



		Predicted	
		Claim	No Claim
Actual	Claim	1971	1571
	No Claim	2269	19385

(a) Resampling - F1 threshold(0.332)

		Predicted	
		Claim	No Claim
Actual	Claim	1122	2420
	No Claim	503	21151

(b) Resampling - F0.5 threshold(0.592)

		Predicted	
		Claim	No Claim
Actual	Claim	1768	1774
	No Claim	1611	20043

(c) Without resampling - F1 threshold(0.292)

		Predicted	
		Claim	No Claim
Actual	Claim	1234	2308
	No Claim	616	21038

(d) Without resampling - F0.5 threshold(0.443)

Figure 5.2: Confusion matrices for the XGBoost model (including ***Yearly + Time variables***) with and without resampling applying thresholds maximizing the F1-score and F0.5-score

Choosing the threshold is facing the trade-off between false positives (type 1 error) and false negatives (type 2 error). Essentially, it comes down to if one wants to be certain that customers predicted with $Claim = 0$ will actually have no claims, thus a high precision, or that the customers with $Claim = 1$ actually are going to have a claim – hence a high recall and negative predicted value. Cancer classification is a typical example of where one would prefer a low false negative, thus aiming for a high recall rate. Applying this logic for insurance, one would want to identify as many customers with claims as possible. However, Figure 5.2 (a) showcases that for the resampled model that this will yield more false positives than true positives. Consequently, it makes more sense for an insurance company to value both precision and recall, with a slight edge to precision. If the latter is in fact most important, the threshold needs to be increased, compared to optimizing for the F1-score, in order to apply the model in an experimental setting. In an experimental setting looking at profit, the threshold would essentially, to an extent, reflect the cost of taking on an unprofitable customer vs. the cost of losing a profitable customer.

Both models (with and without resampling) show similar results and tendencies as the threshold increases. For both thresholds, the resampled model values recall more, in the trade-off between precision and recall, than the model without resampling. The latter model scores best in terms of F1 and F0.5 score for both tresholds, in addition to having the highest AUC score. Meaning that for predicting *Claim*, the XGBoost without resampling is deemed the better model. Even though the observations are not resampled in the training process, the dataset imbalance is addressed by using AUC and F-scores as our comparison metrics, rather than accuracy.

5.1.3 Variable Importance

One of the trade-offs with machine learning models compared to conventional statistical methods is interpretability (Alpaydin, 2020). However, there are some options to provide some insights into the importance of the explanatory variables. The tree-based methods can provide variable importance. The neural network can also theoretically be visualized in terms of variable importance, but due to its complexity it is often not that insightful to interpret (Goodfellow et al., 2016).

The variable importance for XGBoost is displayed in Figure 5.3 and 5.4, for the input-groups ***Yearly variables*** and ***Yearly + Time variables***, respectively. The figures illustrate that the variable importance for the two models is very similar, with the biggest difference being that *Prior Three Years Average Number Of Claims* is included for ***Yearly + Time variables***. This variable was feature engineered and is among the top five most important features for the model. The XGBoost model including time variables weights insurance premium as more important. As there is only a marginal difference in the ROC AUC score when using XGBoost on ***Yearly variables*** vs. ***Yearly + Time variables***, it makes sense that there are no vast differences in the variable importance. Both models have *Has claim last three years prior to Frende* in their top five most important features, indicating that previous records are relevant.

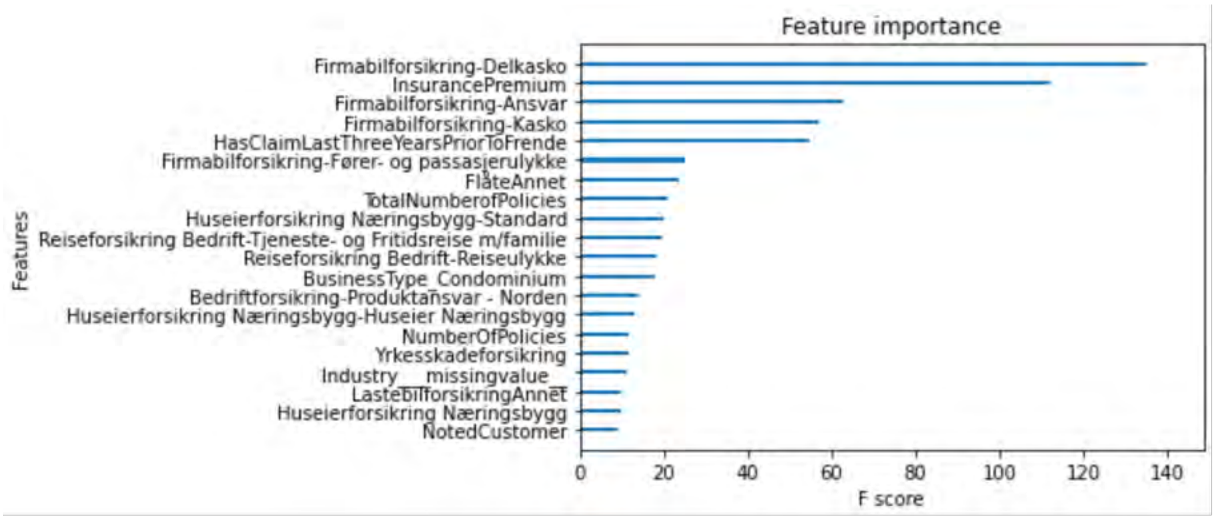


Figure 5.3: Variable importance for XGBoost with ***Yearly variables*** and no resampling

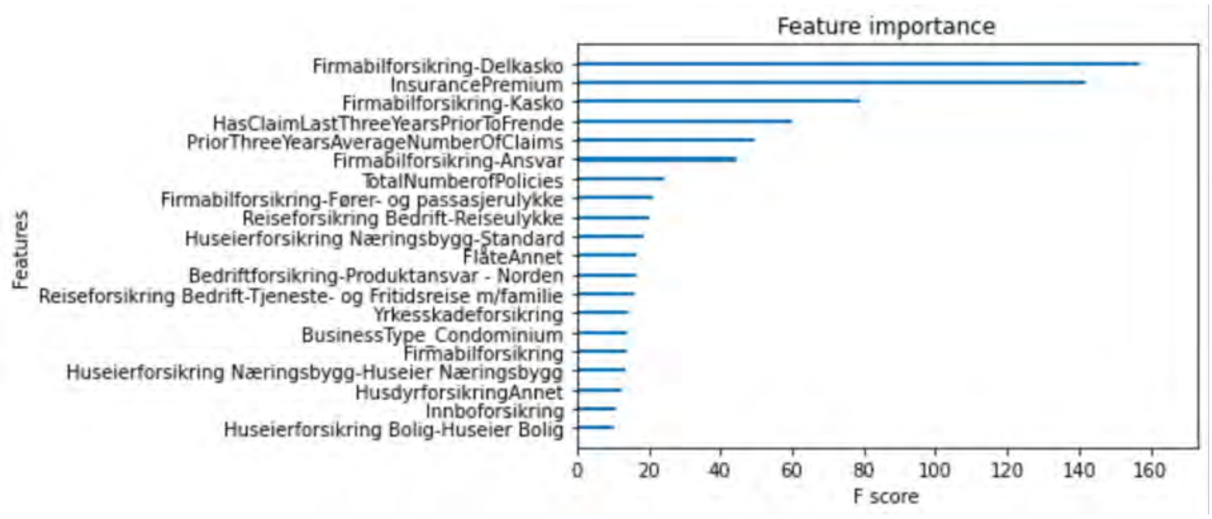


Figure 5.4: Variable importance for XGBoost with ***Yearly + Time variables*** and no resampling

In total we see that it is the data related to insurance types, rather than what kind of business the customer is, that is affecting the probability of claim. *Firmabilforsikring – delkasko*, being a car insurance with partial casco, is the most important feature. From the descriptive statistics of insurance products in Section 4.1.2.2 we observed that vehicular insurance products were represented in terms of most frequent products, total observations with claims and with relatively high shares of *Claim* among its related observations. It makes sense that there exist a high probability of claims having this type of insurance product, thus it is most likely included in the pricing of the product (*Insurance premium*). The severity might, however, not be extreme, meaning that these customers can still be profitable. The explanatory power of company-specific data apart from its chosen products and historic records, seems to be of a very limited value.

5.2 Claim Size

Predicting the *Claim size* if one or more claims takes place is a continuous problem, thus, regression models are used. We will represent the RMSE and MAE metrics for the model from evaluating on the validation set.

5.2.1 RMSE & MAE Comparison

The results from the validation set in terms of RMSE and MAE is displayed in Table 5.5 and 5.6, respectively. All models are performing unsuccessfully with high RMSE and MAE errors. The algorithms are performing better than predicting the mean and/or median of the training set in terms of RMSE, indicating that *Insurance Premium* have some explanatory power. However, more surprisingly the models are performing poorer, or only with marginal improvements, when including the additional input groups – both ***Yearly variables*** and ***Yearly + Time variables*** in terms of RMSE. This indicates that the company-specific variables are not of importance, and that the number of irrelevant variables is in this case lowering the performance of the algorithms, possibly as a consequence of overfitting.

Model	Yearly Premium	Yearly variables	Yearly + Time variables
Mean[1]	323148	323148	323148
Median[2]	328065	328065	328065
GLM inverse gaussian	273076	276200	287941
Regression Tree	290146	296348	291433
Random Forest	290603	283243	282697
XGBoost	<u>271725</u>	283386	280618
Neural network	322648	322166	321446

Table 5.5: RMSE model comparison

Model	Yearly Premium	Yearly variables	Yearly + Time variables
Mean[1]	91516	91516	91516
Median[2]	67518	67518	67518
GLM inverse gaussian	86377	83252	83951
Regression Tree	84001	83468	85994
Random Forest	87552	83252	83951
XGBoost	80299	79634	79993
Neural Network	77096	75871	77718

Table 5.6: MAE model comparison

There seems to be a trade-off between RMSE and MAE. Several of the models with the highest RMSE, have the lowest MAE. Table 5.7, displaying the summary statistics for the observations and the various predictions performed by the models with ***Yearly premium*** as input group, can provide some insight to this.

	Min.	1st Qu.	Median	3rd Qu.	Max.
Observed	148	6300	18575	55482	10079260
GLM inverse gaussian	42	34397	58829	92949	6338867
Regression Tree	480	17401	23774	36178	6693063
Random Forest	7321	33204	55312	87579	6489996
XGBoost	3335	42158	53945	86492	4714608
Neural Network	52677	52757	52921	53223	127783

Table 5.7: Summary statistics of model predictions

From the summary statistics section we see that there is a large variation in the top 5-10% of the observed values of *Claim size*, and we clearly see this within the validation set with its values ranging from 148 to 10 079 260. The models seem to be struggling with finding patterns to explain the variation. This is especially true for the Neural Network which is for more than 75% of the cases predicting the claim sizes to be within the small range of 52 000 – 54 000. XGBoost seems to capture more of the variance, but for the majority of the observations it seems to overestimate the claim sizes. This also applies for GLM, Regression Tree and Random Forest which have larger max values, indicating that there is a larger variation in their predictions.

As seen in Table 5.6, the Neural Network model is the best performing algorithm when considering MAE. This can be explained by RMSE penalizing major deviations more than

[1] Always predict the mean of the training set

[2] Always predict the median of the training set

MAE. When the Neural Network model is always predicting relatively small claim sizes ($<$ 130 000), it will on average get relatively small absolute errors. The infrequent and more extreme values influence MAE substantially less than RMSE as MAE is a linear error function. This is also why predicting the median of the training set minimizes MAE more than all the complex algorithms. However, predicting the same value, or close to the same value, for each observation is suboptimal. Thus RMSE, which seems to lead models to have larger spans in their predictions, looks to be a more suitable error metric in this case.

The various algorithms are performing with similar errors in terms of RMSE. XGBoost is performing best on the ***Yearly Premium*** having a RMSE of 271 725, with the GLM model following very closely. When including ***Yearly variables*** GLM is the best performing model, and moreover when taking the additional time variables into account Random Forest have the lowest RMSE score. Neural Network is as expected the worst performing in regards to RMSE . The Regression Tree is outperforming Random Forest on the ***Yearly premium***, similar to classification tree for with the same input group, but have the highest RMSE error of all three-based methods for the other input-groups.

In total we observe that the models are struggling with predicting the severity of the claims. All models are seemingly weak, but out of the group the best performing model on the validation set in terms of RMSE is the XGBoost model with ***Yearly Premium*** as input group. One should not put too much emphasis on this, as XGBoost performing the best could be random, when the differences among the models are relatively small and seems slightly arbitrary. In addition, the XGBoost model only yields a marginally lower RMSE error than GLM.

6 Experimental Implementation

6.1 Combined Model

The final model combines the classification model predicting *Claim* and the regression model predicting *Claim Size* for more experimental purposes, with the goal of predicting a customers total expected cost, as follows:

$$Total\,expected\,cost = Claim * Claim\,size \tag{6.1}$$

From Claim Occurrences (Section 5.1) and Claim Size (Section 5.2), we selected XGBoost for both classification (without resampling) and regression, with the input-groups ***Year + Time variables*** and ***Yearly premium***, respectively. Meaning that the combined model is the following:

$$Combined\,model = XGBoost\,(Claim) * XGBoost\,(Claim\,size) \tag{6.2}$$

As can be observed in chapter 5, modelling *Claim* seems to be more successful than predicting the severity if such a claim takes place. The error metrics RMSE and MAE are useful for comparing the various algorithms and input-groups when predicting *Claim size.* However, it can be hard to grasp how well the model performs for its purpose, which is to identify profitable corporate customers by predicting *Total expected cost.* To assess the combined model, we will introduce results from a more experimental application of the model on the test set which is the data from 2021.

Using Frende's definition of a profitable customer (defined in the Section 2.1), we can label all corporate customers with a total expected cost larger than 70%*Insurance premium as "Not profitable".

For experimental purposes we can cut all observations that are labeled as "Not profitable" from Frende's customer portfolio for their respective year. In reality, one would not drop a customer that is expected to be unprofitable, but rather adjust the premium accordingly. Adjusting the premium, would, however, depend on the regression model performing better, such that it can facilitate prices that are not possibly obscure. The simplified experimental version, dropping the customers, will still provide some insight as to the model's ability to identify good or bad corporate customers.

6.2 Optimal Threshold for Claim in Terms of Profit

Predicting *Claim*, we observed that the threshold for classifying the predictions greatly affected the rates of false positives and false negatives in terms of having a claim or not. There are three considerations related to the combined model that makes this threshold influential and complicated; (1) True positives can be identified in *Claim*, and as a consequence of a poorly performing regression model the claim severity might be overestimated. From the descriptive statistics we observed that the majority of customers with claims are still profitable. However, if the severity is wrongly predicted to be more than 70% of the insurance premium, the customer is falsely labeled as "Not profitable". Hence, if applying the combined model, one would be better off not identifying this customer as going to have a claim. (2) The prediction of *Claim* can also yield a large amount of false positives. False positives can be predicted to have a claim severity larger than 70% of the insurance premium, classifying the customers wrongly as "Not profitable". (3) If false positives were predicted to have low severities, the wrong labelling in the classification model would ultimately not affect the labelling of "Profitable" vs. "Not profitable".

In total, the threshold for *Claim* is important for the combined model and is increasingly important as the regression model is subpar. The optimal threshold in terms of profit is the threshold at which the prediction of *Claim* is such that, when applying the combined model, the profitability is maximized in terms of not letting go of too many good customers and not taking on too many bad customers.

An insurance customer with a claim size higher than 70% of its paid insurance premium is defined as not profitable. Essentially in terms of profits, a simplification is to represent the remaining 30% of the insurance premium as other costs, which can be administrative, but also a too small margin to be worth the risk. Furthermore, we can use this to define a very simplified profit equation for each customer showed in Equation 6.3. The customer portifolios total profits is defined in Equation 6.4.

$$\begin{aligned} Profit &= Insurance\,premium\ -\ Other\ costs\ -\ Total\ cost \\ where\ \ Other\ Costs &= Insurance\ Premium * 30\% \end{aligned} \tag{6.3}$$

$$Total\ profit = \sum Insurance\,premium\ * 70\%\ -\ \sum Total\ cost \tag{6.4}$$

The relationship between total actual profits and thresholds for *Claim* is displayed in Figure 6.1 for the XGBoost model with ***Year + Time Variables*** as input group using the validation set. The optimal threshold for classifying *Claim* in terms of profits on the validation set is *0.8*. This threshold is significantly higher than the threshold maximizing the F0.5 score, found in the training process of *Claim*, of 0.592. Nevertheless, in terms of profit, the high threshold indicates that the precision is even more favored compared to recall, than what the F0.5 score allows for.

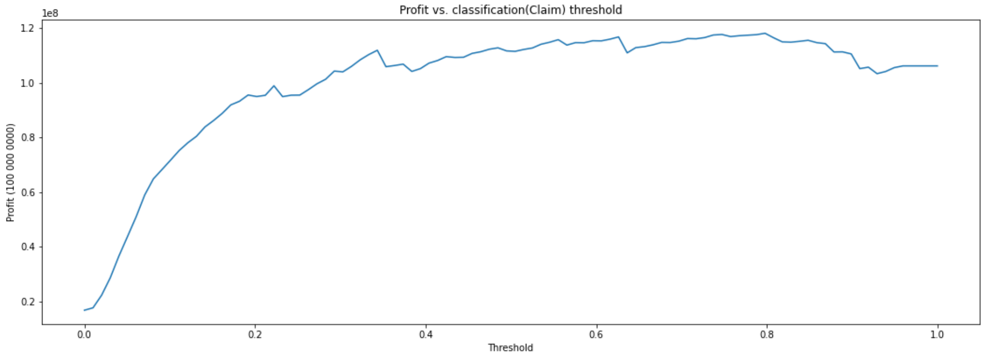


Figure 6.1: Profit in validation set vs. threshold when classifying *Claim*

6.3 Confusion Matrix of Profitable Customers

The confusion matrix in Figure 6.2 shows the relationship between the predicted profitability labels (applying 0.8 as a threshold for *Claim*) and the true labels in the validation set. Among the predicted “Not profitable” there is a precision of 27.16 %, and a recall of 7.93%. Thus, as expected, we see that precision is more important than recall, such that one does not misclassify too many of the actual profitable customers. Both metrics are low, suggesting that the optimal solution based on this combined model, is struggling with the identification of “Not profitable” customers.

		Predicted	
		Profitable	Not profitable
Actual	Profitable	23132	362
	Not profitable	1567	135

Figure 6.2: Confusion matrix of predicted vs. actual profitable and not profitable customers in the validation set

Cutting the predicted unprofitable customers would essentially result in dropping 497 of the corporate customers in the validation set. This includes 135 actual unprofitable customers, and 362 customer that would have been profitable as shown in Figure 6.2. Meaning, that in terms of the trade-off between the gain of dropping bad customers vs. loosing good customers, in general, the cost of one bad customer is higher than the income gained from one good customer. Otherwise, the optimal threshold would have been close to 1, indicating that no customer should be labeled with “Claim”, as it could lead them to being labeled as unprofitable (if the expected claim severity were relatively high). The recall rate for “Not profitable” would have been higher if a lower threshold was applied. However, in terms of profits it would have been too expensive having a larger group of falsely labeled “Not profitable”.

To get a less biased result of how well the model is able to identify profitable and unprofitable customers, this needs to be further investigated using the test-set.

6.4 Alternative Profitability in 2021

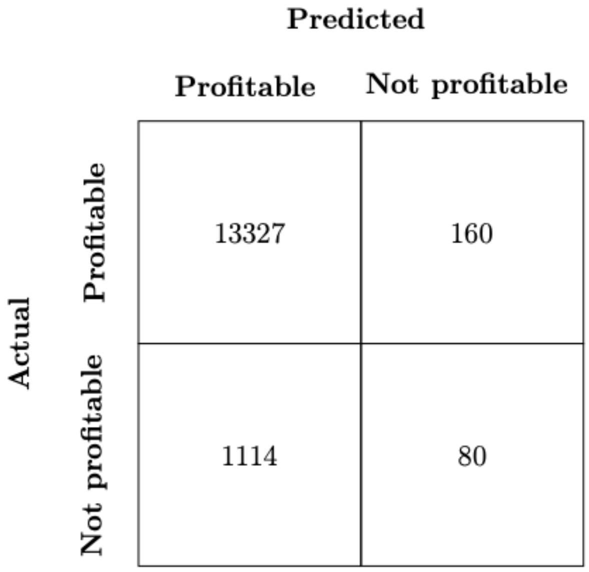
Combing the model with the newly defined optimal threshold (0.8) for binary classification we can use the predictions to cut unprofitable customers for 2021. The XGBoost combined model is essentially competing against what Frende is currently doing. Consequently, the company's earnings is derived by applying Equation 6.4 on the corporate customer portfolio for 2021. Table 6.1 shows what happened in 2021 vs. the alternative truth which essentially is cutting the customers the combined model predicts to be unprofitable.

	Model 2021	Frende 2021	Net effect (Frende – model)
Sum insurance premium (70%)	209 066 136	233 991 693	-24 925 557
Sum paid out claims	163 644 082	190 660 641	+27 016 559
Total profits	45 422 053	43 331 052	+2 091 001

Table 6.1: Profitability in 2021 vs. alternative profitability applying the XGBoost model

There is, as expected from our model, a drop in both the sum of insurance premiums and claim sizes. The net effect is an increase in profits of 2 091 001 NOK, which is a relative increase in profits of 4.8%.

From the confusion matrix in Figure 6.3 we see that Frende would have dropped 80 unprofitable customers, but also losing the premiums of 160 customers that would not have required any pay-outs. In total, the model is able to identify some unprofitable customers, and at the same time not let go of too many good customers when doing so, as we get a net positive result. However, 93% of the unprofitable customers were not identified.



Actual \ Predicted	Profitable	Not profitable
Profitable	13327	160
Not profitable	1114	80

Figure 6.3: Confusion matrix of predicted vs actual profitable and not profitable customers for 2021

In the customer portifolio of 2021 8.85% of the customers are unprofitable, whilst the precision among the 240 observations labeled “Not profitable” is 33.33%. This essentially indicates that the XGBoost combined model has a substantial better performance in identifying unprofitable customers, than if one were to draw 240 observations randomly.

The results are interesting, but there should not be put too much emphasis on this model assessment. It is an experimental application of the combined model with a considerable simplification of the insurance business. The regression model is evaluated to be performing relatively poorly, thus this will greatly affect the combined model. In addition, there is a significant amount of false negatives and false positives in the classification model.

7 Discussion & Conclusion

7.1 Discussion

The modelling of *Claim* returned, with the inclusion of all available variables, relatively satisafactory results for several models. XGBoost was the best performing model of the machine learning algorithms and outperformed the GLM benchmark model. This is in line with a lot of previous work in the insurance field, highlighting the benefits of the more advanced tree-based methods. However, despite the fact that several studies(referred to in Section 2.2) show promising results for using Neural Networks in the context of ratemaking, the Neural Network model for claim occurrences was not as promising. It is important to note, however, that the potential for further optimization of the deep-learning model could potentially improve the results. There are many architecture and tuning efforts to apply, and the ones used in this method are not necessarily optimal.

Additionally, the modelling of *Claim* also touched upon resampling methods. The applied combination of SMOTE and undersampling did not yield any gain, but rather marginally inferior performance for the algorithms compared to the models without resampling. Nevertheless, resampling could still be of significant value as the resampling strategy of this thesis is constant and there are further sampling methods to explore.

Moreover, the modelling of *Claim* investigated whether aggregated data could provide more explanatory value than solely predicting *Claim* on the basis of yearly premium. The results show that there was a relatively large increase in the models' performance when introducing ***Yearly variables***, but only a marginal improvement after including the time variables. The variable importance shows that the explanatory power of company-specific data, apart from its chosen products and historical records, seems to be of limited value. Meaning that company characteristics such as the number of employees, company age, location and the operational industry do not seem to have a significant effect on how likely the customer is to make a claim for the upcoming year.

The ability of predicting the severity of claims proved difficult as showed in the results from the *Claim size* predictions. The aggregated data did not seem to provide any explanatory power. The inclusion of additional variables, compared to predicting solely using the insurance premium as the input variable, rather seemed to contribute with noise worsening the performance. The weak regression models essentially reflects that the identification of profitable/unprofitable customers is limited. The models are doing a reasonably good job identifying customers that will have claims, but are struggling with predicting the claim sizes of these incidents.

The experimental applied approach of the combined model is limited with regards to its simplicity and the errors in both the classification and regression model. In reality, Frende would not cut a customer. The insurance premiums would rather be adjusted to a size in which the corporate customer would be expected to be profitable. Essentially setting a too high price for customers would potentially lead Frende to lose customers to competitors. Additionally, we must add to the discussion that the pricing and acquisition of customers, is highly affected by internal politics and strategy.

The findings from the experimental approach with respect to profitability is still interesting. It shows that the data provided contained information enough to increase profitability for Frende in 2021 by applying their "rule of thumb" definition of profitability and subsequent profits. Information with regards to insurance products should already be represented in the insurance premiums, and we have observed that the most important features are these variables. Thus, our results could also indicate that there exist some potential for improvement as to how the individual insurance premiums are set. The potential for improvement could then, of course also be a result of internal politics, strategy and volume discounts.

Another interesting point the modelling touched upon, is the trade-off between false positive and false negatives. The findings from the experimental approach showed that in order for the business itself to be most profitable, one cannot be too strict in excluding possibly unprofitable customers. The total profitability of the customer portfolio will be highest if one takes a substantial amount of risk. Thereby, knowing that several of the companies in the customer base will be unprofitable. This is due to the alternative cost of not receiving the premiums from the profitable wrongly labeled customers is higher.

7.2 Conclusion

This thesis aimed to provide insights to the usefulness of aggregated data in predicting *Claim* and *Claim size*, thus also predicting a customer's expected cost and finally identify profitable customers through an experimental setting. The results showed that the aggregated data did not add any additional information to insurance premium in terms of predicting *Claim size*, but rather contributed to worsen the predictions in terms of RMSE. For predicting *Claim* the aggregated data did improve the AUC substantially through a binary classification of *Claim*.

Aggregated data improved the models' performance, but the company-specific characteristics were not deemed important in terms of variable importance. The insurance products and historical records were, together with insurance premium, the most important. Ultimately, through the experimental setup, the models were partly able to identify profitable customers by identifying several of the customers with claim occurrences, although the identification of profitable customers was limited mainly due to the difficulties with predicting the associated severities.

To conclude, this thesis provides a small step in disrupting the actuarial comfort zone of ”business as usual”. The results support the several studies arguing the benefits of tree-based models compared to the generalized linear models. XGBoost was the best achieving model for both predicting *Claim* and *Claim size*. Even though the implemented Neural Networks were outperformed by several of the tree-based methods, the flexibility offered by deep-learning models make them highly relevant, and with many more architectures and tuning efforts to explore. Regardless, this thesis shows that both tree-based models and Neural Networks can improve upon the predictive capabilities offered by Generalized Linear Models.

7.3 Further Research

The thesis utilizes in total 45 different models through five algorithms, three input-groups and resampling in the classification case. Essentially this means that the thesis aims to investigate a wide spectrum of responses and machine learning models. This is done to answer the research questions related to the predictive capabilities of machine learning methods on aggregated data. The drawback of this is the need to make restrictions on optimization of the models, in order to answer the comprehensive problem statement within a temporal and computational budget.

A limitation is the applied data, both in terms of feature engineered variables and the available data retrieved from Frende. Aggregated data improved the models' performance in the classification case, but the company characteristics were not deemed important. Including more and different types of company characteristics, the conclusion might be that some companies based on their characteristics do in fact have a higher likelihood of claim and larger claim sizes. It would especially be interesting to include financial data such as liquidity and annual results for the companies. We have experimented with including financial data from Proff (Norwegian database), but the resulting data had close to 60% missing values among several metrics. This would have demanded substantial work in terms of both time and computational effort. However, it is definitely a path that is worth exploring.

The study could potentially benefit from applying more advanced imputation methods, especially regarding *Company age* and the feature engineered time variables. This thesis applied imputation mainly through KNN or using the median of the none missing values. However, more advanced multiple imputation methods could be applied, such as using Random Forest as the imputation estimator. This could generate substitute values closer to the true values of the missing data. As discussed in the thesis, the time variables only contributed to a modest performance improvement for the classification models and yielding higher RMSE for the regression models. These results could be related to historical records not being of substantial importance, or it could also be a consequence of the current imputation causing significant bias.

This study touched upon several machine learning techniques, but there exist a vast of opportunities related to including more machine learning algorithms and improving the current models. Models such as LightGMB, Support Vector Machines or Weighted Random Forests are models worth exploring further. To introduce time series analysis into the predictions, one could explore models such as Recurrent Neural Networks. Furthermore, the performance of the current models are subject to further improvement through additional tuning efforts as the hyperparameter are tuned with somewhat limited grid searches. We particularly suspect large possibilities for improvement in terms of the resampled models and the Neural Networks. The resampling was, in this study, limited due to a constraint on computational resources, and we believe that there could be additional gain through different forms of sampling methods and parameters. Neural Networks offer great flexibility, but are rather complex in terms of tuning. Thus, model performance might increase by taking additional advantage of this flexibility.

There are several limitations to the combined model and the chosen responses in this thesis. As seen in the descriptive chapter (Section 4.1.2, the claims often consist of one or two claims. If an observation has two claims, x and y, the data only represent the total claim amount z, which is the sum of the severities of claims x and y. However, an alternative is to estimate the *Average claim size* ($z/2$ in this case). Subsequently, rather than a binary classification of *Claim*, a different approach with a continuous prediction representing the number of claims (*Approved claims*) should be predicted resulting in:

$$Total\,expected\,costs\; = Approved\,claims * Average\,claim\,size$$

We have experimented with these responses for some of our models, but without any improvements. However, we suspect there still to be a potential in modelling these responses, both as stand-alone models and to be combined as a prediction of the total expected costs.

As discussed thoroughly in this thesis, the evaluation of the combined model is notably experimental with some limitations. Firstly, the simplicity of cutting expected unprofitable customers does not necessarily reflect actual practice as the usual approach would be to adjust the premiums. Moreover, the definition of when a customer is deemed profitable is significantly impactful for the results. By exploring additional profit definitions, one might enhance the evaluation and bring light to new and interesting nuances.

References

Agarap, A. F. (2018). Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375.*

Alpaydin, E. (2020). *Introduction to machine learning.* MIT press.

Bärtl, M. and Krummaker, S. (2020). Prediction of claims in export credit finance: A comparison of four machine learning techniques. *Risks*, 8(1):22.

Beretta, L. and Santaniello, A. (2016). Nearest neighbor imputation algorithms: a critical evaluation. *BMC medical informatics and decision making*, 16(3):197–208.

Blier-Wong, C., Cossette, H., Lamontagne, L., and Marceau, E. (2020). Machine learning in p&c insurance: A review for pricing and reserving. *Risks*, 9(1):4.

Boodhun, N. and Jayabalan, M. (2018). Risk prediction in life insurance industry using supervised learning algorithms. *Complex & Intelligent Systems*, 4(2):145–154.

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

Brownlee, J. (2016). *Machine learning mastery with Python: understand your data, create accurate models, and work projects end-to-end.* Machine Learning Mastery.

Brownlee, J. (2020a). *Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python.* Machine Learning Mastery.

Brownlee, J. (2020b). *Imbalanced classification with python: Better metrics, balance skewed classes, cost-sensitive learning.* Machine Learning Mastery.

Chai, T. and 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.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). Special issue on learning from imbalanced data sets. *ACM SIGKDD explorations newsletter*, 6(1):1–6.

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Chollet, F. (2021). *Deep learning with Python.* Simon and Schuster.

Clarke, R. and Libarikian, A. (2014). Unleashing the value of advanced analytics in insurance. *McKinsey & Company, Aug.*

Dobson, A. J. and Barnett, A. G. (2018). *An introduction to generalized linear models.* Chapman and Hall/CRC.

Donders, A. R. T., Van Der Heijden, G. J., Stijnen, T., and Moons, K. G. (2006). A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091.

Dugas, C., Bengio, Y., Chapados, N., Vincent, P., Denoncourt, G., and Fournier, C. (2003). Statistical learning algorithms applied to automobile insurance ratemaking. In *CAS Forum*, volume 1, pages 179–214. Citeseer.

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., and Herrera, F. (2018). *Learning from imbalanced data sets*, volume 10. Springer.

Frees, E. W., Derrig, R. A., and Meyers, G. (2014). *Predictive modeling applications in actuarial science*, volume 1. Cambridge University Press.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

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

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

Guelman, L. (2012). Gradient boosting trees for auto insurance loss cost modeling and prediction. *Expert Syst. Appl.*, 39:3659–3667.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications*, 73:220–239.

Hanafy, M. and Ming, R. (2021). Machine learning approaches for auto insurance big data. *Risks*, 9(2):42.

Harrison, M. (2019). *Machine Learning Pocket Reference: Working with Structured Data in Python.* O'Reilly Media.

Hastie, T., Tibshirani, R., Friedman, J. H., and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

He, H. and Ma, Y. (2013). *Imbalanced learning: foundations, algorithms, and applications.* Wiley-IEEE Press.

Hossin, M. and Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2):1.

Huang, J. and Ling, C. X. (2005). Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering*, 17(3):299–310.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

Johnson, J. M. and Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):1–54.

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

Mello, A. (2020). Xgboost: Theory and practice.

Müller, A. C. and Guido, S. (2016). *Introduction to machine learning with Python: a guide for data scientists.* " O'Reilly Media, Inc.".

Natekin, A. and Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21.

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

OECD (2020a). Gross insurance premiums (indicator). Accessed on 9. April 2022.

OECD (2020b). The impact of big data and artificial intelligence (ai) in the insurance sector. Technical report.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Probst, P., Boulesteix, A.-L., and Bischl, B. (2019). Tunability: importance of hyperparameters of machine learning algorithms. *The Journal of Machine Learning Research*, 20(1):1934–1965.

Taylor, R. (1990). Interpretation of the correlation coefficient: a basic review. *Journal of diagnostic medical sonography*, 6(1):35–39.

Trenerry, C. F. (2009). *The origin and early history of insurance: including the contract of bottomry.* The Lawbook Exchange, Ltd.

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

Zheng, A. and Casari, A. (2018). *Feature engineering for machine learning: principles and techniques for data scientists.* " O'Reilly Media, Inc.".

Appendix

A1 Complete Overview of Data

The complete overview of the explanatory variables included in the modelling of both classification and regression models are displayed in Table A1.1. The response variables used in the models are displayed in Table A1.2.

Variable	Description	**Type**	**Yearly premium**	**Yearly variables**	**Yearly + time variables**
Insurance premium	Premium paid to Frende for its products	Nummeric	x	x	x
Year		Nummeric		x	x
Total number of policies	Sum insurance products	Nummeric		x	x
Policy years	Sum policies years. Per policy;1 if the insurance is held in a whole year, 0.5 for 6 month etc.	Numeric		x	x
Business type		Categoric		x	x
Distribution channel	Where the customer was obtained	Categoric		x	x
County		Categoric		x	x
Number Of Employees		Nummeric		x	x
Defined Number Of Employees		Binary		x	x
Noted customer	1 if abnormal claim size or claim frequency, and/or missing payments, 0 otherwise	Binary		x	x
Credit score	Scored from 1-5 when becoming a Frende costumer	Categoric		x	x
Business Address Land Code	Two letter country code	Categoric		x	x
Bankrupt	1 if bankrupt, 0 otherwise	Binary		x	x
Under settlement	1 if under settlement (bankrupt), 0 otherwise	Binary		x	x
Insurances	412 types of insurance products, each as a variable with how many polices company x has of product y	Multiple numeric variables		x	x
Has Claim Last Three Years Prior To Frende	1 is claim prior to being a customer, 0 otherwise	Binary		x	x
Main Industrial Classification Category	A-U, NACE	Categoric		x	x
Company age	Year - foundation date.	Nummeric		x	x
Customer length	Number of years as a Frende- customer	Nummeric		x	x
Number Of Policies t-1		Nummeric			x
Approved Claims t-1		Nummeric			x
Claim Size t-1		Nummeric			x
Claim Frequency t-1		Nummeric			x
Claim Percentage t-1		Nummeric			x
Number Of Policies t-2		Nummeric			x
Approved Claims t-2		Nummeric			x
Claim Size t-2		Nummeric			x
Claim Frequency t-2		Nummeric			x
Claim Percentage t-2		Nummeric			x
Prior Three Years Average Claim Frequency		Nummeric			x
Prior Three Years Average Claim Percentage		Nummeric			x
Prior Three Years Probability Of Claim		Nummeric			x
Prior Three Years Average Number Of Policies		Nummeric			x
Prior Three Years Average Number Of Claims		Nummeric			x
Prior Three Years Average Claim Size	Only included claim sizes > 0 when averaging	Nummeric			x
					x
Delta Number Of Policies y-1	Number of policies today – last year	Nummeric			x
Delta Number Of Policies y-2	Number of policies today – two years ago	Nummeric			x

Table A1.1: Complete overview of all explanatory variables included in predicting *Claim* and *Claim size*

Reponse variable	Description	Data type
Claim size	Total amount paid out to the company, can consist of more than one claim. 0 when not defined.	Nummeric
Claim	1 approved claims> 0, 0 otherwise	Binary

Table A1.2: Response variables for the predictive regression and classification models

A2 Complete Overview of All Included Insurance Products

Table A2.1 displays all 130 included products as nummeric variables holding the amount of policies of the respective insurance product that the customer i holds in year t.

Insurance products	Insurance products	Insurance products
UlykkesforsikringAnnet	Fritidsbåtforsikring	Arbeidsmaskinforsikring-Fører- og passasjerulykke
LandbruksforsikringAnnet	Gruppeliv eierbank	Arbeidsmaskinforsikring-Kasko 1.risiko
DyreforsikringAnnet	Hesteforsikring	Bedriftforsikring-Ambulerende verktøy
HytteforsikringAnnet	Huseierforsikring Bolig	Bedriftforsikring-Avbruddstap fullverdi
CampingvognforsikringAnnet	Huseierforsikring Næringsbygg	Bedriftforsikring-Bedriftsansvar - Norden
BilforskringAnnet	Husforsikring	Bedriftforsikring-Kasko
Kollektiv ulykkesforsikringAnnet	Hytteforsikring	Bedriftforsikring-Naturskade
FirmabilforsikringAnnet	Innboforsikring	Bedriftforsikring-Panthaver
BedriftsforsikringAnnet	Kollektiv ulykkesforsikring	Bedriftforsikring-Produktansvar - Norden
CyberforsikringAnnet	Lastebilforsikring	Bedriftforsikring-Tingdekning på fast forsikringssted
BrannforsikringAnnet	Maskinforsikring	Firmabilforsikring-Ansvar
FlåteAnnet	Mopedforsikring	Firmabilforsikring-Delkasko
GruppelivAnnet	Motorsykkelforsikring	Firmabilforsikring-Fører- og passasjerulykke
InnboforsikringAnnet	Prosjektforsikring	Firmabilforsikring-Kasko
Huseierforsikring NæringsbyggAnnet	Prøveskiltforsikring	Firmabilforsikring-Leasing
Huseierforsikring BoligAnnet	Reiseforsikring Bedrift	Firmabilforsikring-Leiebil
HesteforsikringAnnet	Tilhengerforsikring Bedrift	Firmabilforsikring-Panthaver
FritidsbåtforsikringAnnet	Tilhengerforsikring	Huseierforsikring Bolig-Naturskade
AnsvarsforsikringAnnet	Tilleggsnæringforsikring	Huseierforsikring Bolig-Standard
HusdyrforsikringAnnet	Ulykkesforsikring	Huseierforsikring Næringsbygg-Naturskade
MaskinforsikringAnnet	Uregistrerte kjøretøy	Huseierforsikring Næringsbygg-Standard
MopedforsikringAnnet	Verdigjenstandforsikring	Prosjektforsikring-Bygg, anlegg og montasje
MotorsykkelforsikringAnnet	Veterankjøretøyforsikring	Prosjektforsikring-Naturskade
YrkesskadeforsikringAnnet	Yrkesskadeforsikring	Prosjektforsikring-Verktøy og utstyr
Annen sykdom eierbankAnnet	Arbeidsmaskinforsikring-Arbeidsmaskin	Reiseforsikring Bedrift-Ansvar og rettshjelp
FjørfeforsikringAnnet	Bedriftforsikring-Ansvar risiko	Reiseforsikring Bedrift-Avbestilling
EnkeltprosjektforsikringAnnet	Bedriftforsikring-Maskiner/inventar/løsøre/varer	Reiseforsikring Bedrift-Forsinkelse
PrøveskiltforsikringAnnet	Bilforsikring-Bil	Reiseforsikring Bedrift-Reisegods
TilhengerforsikringAnnet	Cyberforsikring-Cyberforsikring	Reiseforsikring Bedrift-Reisesyke
Reise fortsettelsesforsikringAnnet	Firmabilforsikring-Firmabil	Reiseforsikring Bedrift-Reiseulykke
ReiseforsikringAnnet	Huseierforsikring Bolig-Huseier Bolig	Tilhengerforsikring Bedrift-Brann/Tyveri 1.risiko
LastebilforsikringAnnet	Huseierforsikring Næringsbygg-Huseier Næringsbygg	Tilhengerforsikring Bedrift-Kasko 1.risiko
TrumfVisaReiseAnnet	Lastebilforsikring-Lastebil	Tilleggsnæringforsikring-Driftstap Ulykke
Uregistrerte kjøretøyAnnet	Motorsykkelforsikring-Motorsykkel	Verdigjenstandforsikring-Standard
VeterankjøretøyforsikringAnnet	Prosjektforsikring-Bygge-, anleggs- og montasjearbeid	Yrkesskadeforsikring-Fritidsulykke død
ArbeidsmaskinforsikringAnnet	Reiseforsikring Bedrift-Tjeneste- og Fritidsreise m/familie	Yrkesskadeforsikring-Fritidsulykke invaliditet
Ansvarsforsikring	Tilhengerforsikring Bedrift-Person-/Varebiltilhenger	Yrkesskadeforsikring-Fritidsulykke uførhet
Arbeidsmaskinforsikring	Tilleggsnæringforsikring-Tilleggsnæring	Yrkesskadeforsikring-Ulykke v/reise til/fra arbeid
Bedriftforsikring	Verdigjenstandforsikring-Verdigjenstand	Yrkesskadeforsikring-Yrkesinvaliditet under 15Bilforsikring
Yrkesskadeforsikring-Yrkesskade - ansatte	Yrkesskadeforsikring-Yrkesskade Sykdom	
Brannforsikring	Yrkesskadeforsikring-Yrkesskade - selvst. næringsdriv.	Yrkesskadeforsikring-Yrkesskade Sykdom - Lovpålagt
Cyberforsikring	Arbeidsmaskinforsikring-Arbeidsmaskinansvar	Yrkesskadeforsikring-Yrkesskade Ulykke
Firmabilforsikring	Arbeidsmaskinforsikring-Bilansvar	Yrkesskadeforsikring-Yrkesskade Ulykke - Lovpålagt
Arbeidsmaskinforsikring-Brann/Tyveri 1.risiko		

Table A2.1: Overview of all included insurance products

A3 Grid Search for Tuning Hyperparameters

Table A3.1 shows an overview of the initial grid searches when tuning hyperparamteres for the respective models.

Model	**Hyperparameters**	**Grid search**
GLM	Lambda (lasso)	[2, ... , 0.001]
RF	Max depth	[20, 40, ... , 120]
RF	Number of features selected (classification)	[20, 40, 60, 80, $\sqrt{p}$]
RF	Number of features selected (regression)	[20, 40, 60, 80, $\frac{p}{3}$]
RF	Number of trees in the ensemble	[200, 300, ... , 1000]
RF	Min number of samples in each split	[2, 5, 10, 20]
RF	Min number of samples in each leaf	[2, 4, 8, 12]
XGB	Learning rate	[0.001, 0.01, 0.02, 0.03]
XGB	Max depth	[4, 6, 8, 10]
XGB	Min child weight	[1, 3, 7, 10]
XGB	Reg lambda	[0, 10, 20]
XGB	Reg alpha	[5, 8, 10, 15, 20]
XGB	Gamma	[0, 0.5, 1]
XGB	Subsample	[0.5, 0.7, 1]
XGB	Colsample bytree	[0.5, 0.7, 1]
NN	Learning rate	[0.001, 0.01, 0.01]
NN	Layers	[3, 5]
NN	Neurons	[32, 64, 96]

Table A3.1: Initial grid search for tuning hyperparameters

The GLM models have been tuned by looping through possible values for λ i.e., the shrinkage parameter (lasso). CARTs have been tuned in a similar manner, but with the effective alphas returned by scikit-learn's *cost_ complexity_pruning_path* function resulting in roughly 1 000 candidates.

XGBoost and Random Forest was tuned using the grid searches displayed in Table A3.1 in the first round. The parameters were then further tuned in a second round. The second grid search explored the values closer to the returned optimal value for each of the parameters from the initial search. In the case of Random Forest, the second round put emphasis on the *Number of trees* and *Number of features* parameters. This was done as the alternative, expanding the initial grid searches, would have exceeded the computational capacity available.

Neural Networks were tested with some different learning rates, layers and neurons. As deep Neural Networks offer great flexibility, there are a vast of more combinations that could be explored.