Stockholms universitet

Application and Comparison of Machine Learning and Traditional Methods to Insurance Pricing in Scarce Data Environments

Asmir Prepic

Masteruppsats i försäkringsmatematik
Master Thesis in Actuarial Mathematics

Masteruppsats 2021:6
Försäkringsmatematik
Juni 2021

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Stockholm University

Mathematical Statistics
Stockholm University
Master Thesis **2021:6**
http://www.math.su.se

Application and Comparison of Machine Learning and Traditional Methods to Insurance Pricing in Scarce Data Environments

Asmir Prepic*

June 2021

Abstract

Complex models and methods has received plenty attention over the recent years and various authors have shown the power of e.g. neural networks and random forests over traditional insurance pricing models. This thesis investigates the predictive power for a simulated insurance portfolio where there is less exposure among policyholders who have higher risk by utilising a synthetic minority oversampling technique (SMOTE) and comparing the predictive performance without application of SMOTE. In addition the same comparison is applied to a real insurance data set. The thesis shows that without SMOTE and where there is clearly less exposure among high risk customers compared to the rest of the portfolio, the traditional vanilla GLM outperforms the more complex models in predictive power. On the contrary, by utilizing SMOTE and oversampling the high risk policyholders such that the data is more balanced, neural networks, regression trees and random forests make better prediction based on the 10 fold cross validation technique.

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

Contents

1 Introduction

A non-life insurance contracts purpose is to transfer economic risk from the policyholder to the insurance company. Insurance companies rely on the law of large numbers for the sum of a large number of small individual losses which ultimately become predictable on an aggregate level. If the previous statement holds, the premium charge for this transfer of economic risk between the policyholder and the insurance company should be based on the expected value of the loss. This expected value is commonly referred to as the pure premium in the industry and is required from the policyholders for the company to be solvent. As a result of the principle and the increased competition on the insurance market, a portfolio will consist of different types for risk profiles. For instance the reimbursement of a vehicle insurance will vary depending on the vehicles price which in turn gives need of statistical models in the insurance rate making process.

Policyholders needs are reflected by the products offered in the insurance market and can attract different samples of the population. Travel insurance is demanded by potential policy buyers who frequently travel which may have different background and different behaviour than the rest of the population that do not travel as frequently. Likewise someone who wants to transfer risk of vehicle damage might be inclined to purchase an insurance that covers damage to a greater extent the newer or more expensive the vehicle is. Calculating the price of the insurance therefore needs to be done with care with respect to the properties of the insured portfolio rather than assuming that the portfolio is a random sample of the general population.

Rate making has in the past, and still is partly done currently, as with regards to some basic information about the policyholder. The information usually results in a few different pieces of information such as age, geographic location etc. called tariffs which are then summed to obtain a insurance price. Naturally the prices and the tariffs can be set and optimized to give a fair price and margin to all customers regardless of their tariff arguments or for example be optimized to reach an overall portfolio profit. In addition, other parameters can be included not directly related to the risk such as price elasticises, longevity of the policies etc. adding more complexity to the problem of insurance rate making.

With the increased competition between insurance companies and increased demands of self-regulation and capital requirements of the insurance companies it is becoming progressively important for insurers to charge fair premiums for their customers. Insurers need therefore to be more precise in determining the correct price for their portfolio and also be dynamic enough to adjust prices when the underlying distribution of the risk changes.

As stated in [24] the increase of data availability and increase of modern tech-

nology drives the rise of alternatives to traditional actuarial methods in order to meet the demand and the complexity of risk. More specifically the writers say that

> ...We need to evolve, working intelligently with a wide cross-function of skill sets such as data experts, data scientists, economists, statisticians, mathematicians, computer scientists and, so to improve the transformation of data to information to customer insights, behavioural experts. This is a necessary evolution for actuaries and the profession to remain relevant in a high-tech business world.

Additionally, it has been reported by consultancy firm Accenture [2] that insurance companies process and use about 10-15% of the available data creating a large potential for applications of recent techniques in the day to day business. Naturally this does not only apply to pricing but also other areas necessary in running an insurance business such as analysis of risk appetite or claims management. The recent development creates potential for uses of more accurate predictive methods in insurance rate making.

Machine learning methods have been applied and discussed in insurance literature extensively [9] [18] [30]. Although the same methods are applicable within rate making, there are not many examples where they have been applied. Insurance pricing is a regulated field which requires some model transparency. The different types of applicable Machine Learning techniques and models are often to complex to be able to fit the transparency criterion. The standard methods for rate making and pricing within PC insurance is a combination of generalized linear models [23]. Utilizing certain distributional assumption with some linear transformation, GLM's in the standard form do not take into account effects of interactions between the studied variables if not explicitly specified. GLM's have also been extended to account for smooth effects of the explanatory variables such as GAM's. Their power in insurance rate making application stems from ability to capture complex structures in risk of the insurance portfolio as well being interpretable by the actuary. GLM's however require some a priori assumption about the underlying data generating process to be able to fit the data as well as possible. Moreover, actuarial methods has historically been adaptable to regression methods [26], [11] and [6] which makes the problems suitable for other types of estimation methods to the ones of GLM.

This paper aims to investigate how Machine Learning methods compare to traditional actuarial rate making methods when there are complexities such as interactions and non-linearity which are difficult to observe from the observed data. Following the introduction of chapter 1, chapter 2 covers the theoretical background applied in this thesis, chapter 3 presents the data on which the methods are applied, chapter 4 presents the results together with the implementation methods applied which are then commented in chapter 5.

2 Theory

This part describes the modelling theory applied in this thesis. Generalized linear models, generalized additive models, tree based models, neural networks and imbalanced data handling is described in the following subsections.

2.1 Insurance Pricing

A fundamental difference in the insurance industry compared to other industries is that costs do not occur at the time of the writing of the product. Costs occur instead if a policyholder experiences a economic loss reimbursable by the written insurance contract. In that sense, the business of selling insurances is essentially selling future promises to anyone that holds the insurance contract. The seller of the insurance contract does not know when the cost will occur therefore has to rely on historical data and estimations to find an adequate price for the contract. Actuarial estimates often consider the pure premium which is the best estimate of the future liabilities per insured. In effect, natural questions occur such as how much to charge a specific group of customers if they for example have less claims than the rest of the portfolio?

Actuarial estimates of the insurance price often deals with some kind of modelled price because outcomes of the costs of the insurance prices might be subject to large random fluctuations. Additionally, prices need to take into consideration the market level of the premiums to be able to attract customers as well as the insurance company's costs for running the business of risk transferring.

Insurance companies cover the costs by collecting premiums from the insurance buyers in exchange for the transfer of risk from the policyholder to the insurance company. Premiums can be paid in various ways but the insurance company needs to hold capital to be able to cover potential losses and it is not therefore uncommon for premiums to be paid in advance (i.e. prior to the period during which risk is transferred or prior to claim payouts). Defining the pure premium (the premium paid by policyholders to cover risk) utilizes the law of large numbers. Define $Y_1, Y_2, Y_3, ...$ as i.i.d random variables representing the claims paid by the insurance company for a period of 1 year. Define

$$\bar{Y_n} = \frac{1}{n}\sum_{i=1}^{n} Y_i \tag{1}$$

as the average claim amount for the total portfolio for the same time period during which $Y_1, Y_2, Y_3, ...$ are observed. If $Y_1, Y_2, Y_3, ...$ are considered to be i.i.d. they will have the same distribution function F such that

$$F(y) = P[Y_1 \leq y] \tag{2}$$

and the expectation

$$E[Y_1] = \int_0^\infty y dF(y). \tag{3}$$

By the law of large numbers

$$\bar{Y_n} \to E[Y_1] \quad \textbf{as} \quad n \to \infty. \tag{4}$$

[14] From (4), a portfolio has better estimates of the average claims costs as the number of policyholders increases. Hence the estimator from (1) improves and the insurance company could charge a premium which does not have to take into account large variations in the outcome of the claims costs to cover the variance of the cost. A central question from this property is how to distribute the expectation of the average claims costs amongst the policyholders fairly and with respect to the market values of similar with the inclusion of loading's for other costs. The expected claims costs can be further split such that

$$\text{pure premium} = \text{claim frequency} \times \text{claim severity}. \tag{5}$$

To be able to split the premiums between the policyholders, various statistical methods are applied estimate fair prices given policyholders characteristics. Failure to do so may generate price difference in the market which can affect the stability of the insurance company and hence the policyholders economic safety.[29]

A classical approach for non-life insurance has been to model insurance costs as

$$S = Z_1 + Z_2 + ... + Z_N = \sum_{k=1}^{N} Z_k \tag{6}$$

where Z_k are individual claim sizes and N is the number of claims. Both N and Z_k are in (6) stochastic meaning that S is also stochastic. It is of interest to study S or the compounded claims cost in a general setting and to understand the structure and use it in analysis. N in (6) is defined as a discrete random variable which is a integer that can be modelled with Poisson distribution

$$P(N = k) = \exp(-\mu)\frac{\mu^k}{k!} \quad \text{for all} \quad k \in \mathbb{N}_0, \tag{7}$$

where $\mathbb{N}_0 = 0, 1, 2, ...$ $\mu > 0$ is the only parameter in (7) and is the expected number of claims if N is number of claims. It is also necessary to introduce a representation of the proportion of time that the insurance company has been in risk. Usually in insurance the proportion of time is called exposure and is measure in insurance years denoted in this paper as w. Instead of number of claims, a volume scaled property will be used in this thesis. By defining

$$Y = \frac{N}{w} \tag{8}$$

if N is a Poisson distributed random variable and w is the risk exposure then Y becomes the claims frequency and will be modelled in this thesis. If the number of claims are considered on the aggregate level where the insured policyholders are subjected to i.i.d risks proportional to their exposure, the expectation of N scaled with the exposure is then

$$E[N] = w\mu \tag{9}$$

and

$$E[Y] = \mu. \tag{10}$$

Similarly if the number of claims are considered on the aggregate level where the insured policyholders are subjected to i.i.d risks proportional to their exposure, the variance is then

$$Var[N] = w\mu \tag{11}$$

since the property of the Poisson distribution with parameter μ is

$$E[N] = Var(N). \tag{12}$$

The variance of Y is then

$$Var(Y) = \frac{w\mu}{w^2} = \frac{\mu}{w}. \tag{13}$$

A interesting property of (13) is that the variance of the frequency Y is reduced as the exposure w is increased. on a very basic level the quantity of interest to model will be

$$Y = \mu + \epsilon \tag{14}$$

with epsilon being centered with variance μ/w.

2.2 Statistical Learning

Define a collection $\mathbf{X} = (X_1, ..., X_p)$ of observed values with certain characteristics, often denoted predictors, independent variables or features, with a observed variable of interest denoted Y called the response or observed variable which ideally exists for each collection of $(X_1, .., X_p)$. Under the assumption that there is some relationship between Y and $(X_1, ..., X_p)$ denoted as

$$Y = f(\mathbf{X}) + \epsilon \tag{15}$$

where $f()$ in (15) is some function of $(X_1, ..., X_p)$ and ϵ is a random error term which consists of the unexplained part of Y from the function $f()$. Naturally, except from some special circumstances or simulation studies, $f()$ is not known but has to be estimated using a set of approaches called Statistical Learning [16]. There a re mainly two reasons for the purpose of estimating $f()$. Prediction of the response Y or inference about Y using the observed characteristics in $(X_1, ..., X_p)$, becomes, with estimate notation,

$$\hat{Y} = \hat{f}(\mathbf{X}). \tag{16}$$

It is the specification, or parameters, of $\hat{f}()$ and the accuracy of $\hat{Y}$ which is of interest to be estimated or analyzed in statistical learning. For prediction purposes, the estimate of Y denoted as $\hat{Y}$, there are two possible sources of error named as reducible error and irreducible error. Errors that occur from the failure of getting a perfect estimate of $\hat{f}()$ is referred to as the reducible error. Unless $\hat{f}()$ is the correct function, then there is always a possibility to improve the fit of $\hat{f}()$ to Y as a function of $(X_1, ..., X_p)$ hence the error is possible to reduce. The irreducible error of the learning problem stems from ϵ of (15). As ϵ is not used in the estimation result of (16), it will not be estimated. This part usually is considered stochastic and is required to have stochastic properties. To measure both the errors of (16) a squared deviance from the true $f()$ can be defined as

$$\begin{aligned} E(Y - \hat{Y})^2 &= E[f(X) + \epsilon - \hat{f}(X)]^2 \\ E(Y - \hat{Y})^2 &= [f(X) - \hat{f}(X)]^2 - Var(\epsilon). \end{aligned} \tag{17}$$

The first term of (17) measures the error which can be reduced and the second is the error which cannot be reduced. [16]

2.2.1 Estimation

A central problem in statistical learning, or in statistics and machine learning separately, is the estimation of $\hat{f}()$ and there are methods of varying complexity as well as properties such as linearity or non-linearity. Regardless of the methods of choice, the principle is to minimize some kind of fit to the data points $(Y_i, X_{i1}, ...X_{ip})$ where i denotes the i-th observation. Two groups of estimation procedures are defined, on a general level, as parametric and non-parametric methods. By assuming, or experimentally proving, that the observations under study follow some kind of well defined distribution or pre-determined model, the parametric approach can be used to find the relationship of interest. To estimate $\hat{f}()$ using parametric methodology, first a assumption is made for the form of $f()$. For example a linear assumption is

$$f(\mathbf{X}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p \tag{18}$$

and requires only the estimation of the parameters $(\beta_1, \beta_2, ..., \beta_p)$ for which there are several estimation methods such as least squares or maximum likelihood for example. The method is considered parametric since there are a fixed number (p) parameters to estimate and the function is .

On the contrary of parametric methods, non-parametric methods do not require any assumptions of the form of $f()$ prior to estimation and can have a unlimited number of parameters. $f()$ is not required to have a linear form as and can take on any form which is necessary to meet some criterion. The goal in the non-parametric setting is to get the estimate $\hat{f}()$ as close to the observed data points as possible with the requirements that $\hat{f}()$ is to a acceptable degree

neither explains data too well nor too poorly. Plenty of texts on the subject of classifying the different types of approaches has been done over time and new are being produced but these will not be further discussed in this thesis. [20]

Depending of the chosen method, the characteristics of the optimizing target changes somewhat. Estimating a function such as (18) or a non parametric function a measure of how far from the true $f()$ the estimated $\hat{f}()$ has to be defined.

$$\text{Error}(\hat{Y}) = \sum_{i}^{n} w_i(Y_i - \hat{Y}_i)^2 \tag{19}$$

is the an example of the error in general terms which is to be minimized with w_i being the exposure meaning that the error is weighted according to how much experience we have from each observation. Expression (19) is defined such that the deviance of the predicted value $\hat{Y}$ from the outcome Y is squared and summed for all the observation. Depending on what quantity Y is to be predicted, (19) can be adjusted such that instead of the deviance squared an absolute deviance can be used instead for example. A function such as (19) can be used in a parametric or non parametric setting as it is a function of the predicted value $\hat{Y}$.

In the case of estimating with maximum likelihood, a prior assumption needs to be made for a distribution for which a likelihood function exists and is well behaved. With a sample of observed i.i.d. random variables of interest $(Y_1, Y_2, ..., Y_n)$ and with the assumption that the data generating process resulting in the observations $(Y_1, Y_2, ..., Y_n)$ is from a Poisson distribution. Under the assumptions, the probability distribution function of the sample $(Y_1, Y_2, ..., Y_n)$ is a product of the individual probability distributions

$$L(\mu, Y_1, ..., Y_n) = \prod_{i=1}^{n} \frac{e^{-\mu}}{Y_i!}\mu^{Y_i}. \tag{20}$$

Searching for a solution for (20) becomes a problem of finding a value for the parameter μ. A criterion for which values μ can have needs to be specified. The term maximum likelihood stems from the criterion to specify μ such that (20) is maximized. Maximization is done such that the log of (20) is used instead to obtain a easier form to work with (20) becoming

$$\log(L(\mu, Y_1, ..., Y_n)) = -n\mu + \sum_{i=1}^{n} Y_i \log(\mu) - \sum_{i=1}^{n} \log(Y_i!). \tag{21}$$

Taking the derivative of (21) with respect to μ and solving for μ results in the maximum likelihood estimate

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

(22)

which is the mean of the observed values. As mentioned in earlier paragraphs of this section, this estimation procedure is parametric resulting in the one parameter estimated which specifies the Poisson distribution.

Expanding the reasoning about maximum likelihood and parametric estimation of a assumed distribution of the observed quantities $(Y_1, Y_2, ..., Y_n)$ to some a error function such as (19). The mean, or weighted mean, of the squared deviance as specified in (19) for example punishes a deviance from the observation equally and symmetrically around the mean. It is usable if the data is observed or assumed to be symmetric but can result in strange punishments if the data is non-symmetric. Since it is assumed that the response is from a Poisson distribution the error function (19) can be improved to more reflect the distribution of choice. Using reasoning from statistics, it is of interest to find the difference between a model of the observations without imposing any restrictions, i.e. a fully saturated model compared to a model with some restrictions. [19] Since assuming a distribution of the data generating process of the observed values gives a probability distribution, the difference of the likelihood between the two models could be utilized as a measure of difference. In essence this gives a likelihood of observing the values of a model given a set of parameters and the likelihood of observing the values given that each observatory is its own parameter. Scaling the difference gives then

$$D(\boldsymbol{\mu}, Y_1, ..., Y_n) = 2\big(\log(L(\mu_{\text{saturated}}, Y_1, ..., Y_n)) - \log(L(\mu_{\text{partial}}, Y_1, ..., Y_n))\big) \quad (23)$$

where L is the likelihood function and the deviance is compared between the saturated model and a model with parameter μ_{partial}. Minimizing the deviance (23) is equivalent to maximizing the likelihood (22). Inserting the Poisson distribution in (23) gives then that

$$D(\mu, Y_1, ..., Y_n) = 2\sum_{i=1}^{n}\Big(Y_i\log(Y_i) - Y_i - \log(Y_i!) - Y_i\log(\mu_i) + \mu_i + \log(Y_i!)\Big). \quad (24)$$

Simplifying (24) gives then

$$D(\mu, Y_1, ..., Y_n) = \sum_{i=1}^{n}\Big(Y_i\log\Big(\frac{Y_i}{\mu_i}\Big) - (Y_i - \mu_i)\Big). \quad (25)$$

A natural extension of (24) used in this thesis is to weight the deviance by how much exposure the observation carries as well as the adjustment for estimating frequencies rather than the count of claims. Weighting (24) results in the weighted deviance on the form

$$\text{Deviance}(\hat{\mu}) = 2\sum_{i=1}^{n}\left(w_iY_i\log\left(\frac{Y_i}{\hat{\mu}_i}\right) - w_i(Y_i - \hat{\mu}_i)\right) \quad (26)$$

where Y_i is the observed frequency of claims for policyholder or group of policyholders i and w_i the exposure or number of insurance years, for policyholder i.[22] Expression (26) is adjusted such that it does not produce undefined variables by introducing the continuous correction. Since the number of claims $N_i = Y_i w_i$ the extension

$$N_i \log(N_i) = \begin{cases} N_i \log(N_i), & \text{if } N_i > 0 \\ 0 & \text{if } N_i = 0 \end{cases} \tag{27}$$

solves the issue of the log not being defined at 0 and having the limit $-\infty$ as the value approaches 0 from its domain. Note that the maximum likelihood is one approach for minimizing (24) among others. Other solutions will be applied in this thesis when dealing with regression trees and neural networks but with the objective of minimizing the same deviance function although with the correction.

2.3 Generalized Linear Models in Non-Life Insurance Pricing

Generalized linear models (GLM's in short) are considered well suited for insurance rate making as the goal is to estimate a response variable Y which is described by and varies with respect to some characteristics of the policyholders. Defining these characteristics as explanatory variables or features denoted as x_i resembles a lot like a standard linear regression or a general type of curve fitting problem. Given the nature of insurance claims and risk more flexibility needs to be added as the losses are generally positive with non-normal errors. To adjust for these non-standard conditions, generalized linear models use transformation of probability distributions in the exponential dispersion family and transformation of the mean function to be a monotone function of the explanatory variables.

The probability distribution of the exponential dispersion family can be written on the form

$$f_{Y_i}(y_i; \theta_i, \phi) = \exp\left\{ \frac{y_i \theta_i - b(\theta_i)}{\phi / w_i} + c(y_i, \phi, w_i) \right\} \tag{28}$$

where y_i is the the response observation of the data or the variable of interest to predict, θ_i is some parameter related to the distribution of interest, ϕ is the dispersion parameter and w_i is some kind of weight related to the observations. In insurance w_i is generally set to exposure often set to insurance years to adjust for that not all observations has been insured equally long time. $b(\theta_i)$ is a monotonic convex function of the parameter. The probability distribution is completely specified by parameters θ_i and ϕ conditioned on the choice of the function $b()$. $c()$ is a function which is not of importance in the pricing application but can be summarized as the part were the rest of the terms which are not dependent on the quantity of interest are collected. In the Poisson case,

$Y_i!$ would be collected in $c()$ [22]. Generalized linear models are specified by their EDM property as described, a linear predictor $\eta = \boldsymbol{X}\boldsymbol{\beta}$ and a linear link function expressing the expected value of $\boldsymbol{Y}$ given the observations $\boldsymbol{X}$ such that

$$E[\boldsymbol{Y}|\boldsymbol{X}] = \mu = g^{-1}(\boldsymbol{X}\boldsymbol{\beta}). \tag{29}$$

For example for the normal distribution of EDM family is the identity link giving

$$\boldsymbol{X}\boldsymbol{\beta} = \mu. \tag{30}$$

Application to the assumed distributions in this thesis are presented in the next section.

2.3.1 Poisson Distribution for Claims Count

Of the EDM families, the Poisson distribution is commonly used for count data and thus is suitable to model the stochastic process of claims count given certain assumptions which often are satisfied for insurance applications. [25]. If $N(t)$ is defined as the number of claims during the time interval $[0, t]$ and $N(0) = 0$, i.e. that at time 0 no claims have been observed. If then an additional assumption is that the risk of the policies in the portfolio are independent, the Poisson distribution holds for the aggregate portfolio as well. Because insurance policies can start during any time of the year, it can be assumed that the risk is proportional to how long during a period the insurance company has transferred the risk. If X_i is the number of claims for policyholder or group of policyholders with similar risk i. The probability density function of X_i is then defined as

$$f_{X_i}(x_i; \mu_i) = e^{(-w_i\mu_i)}\frac{(w_i\mu_i)^{x_i}}{x_i!}, \qquad i = 1, 2, 3, ... \tag{31}$$

By transforming the number of claims, the claims frequency can be expressed as $Y_i = X_i/w_i$ which then transforms the Poisson distribution to the form

$$f_{Y\,i}(y_i; \mu_i) = P(Y_i = y_i) = P(X_i = w_i y_i) = e^{-w_i\mu_i}\frac{(w_i\mu_i)^{w_i y_i}}{(w_i y_i)!}, \tag{32}$$

with $w_i y_i$ being a positive integer.[22]

2.3.2 Gamma Distribution for Claim Severity

Similarly to the Poisson distribution there are two quantities of interest for the claims cost of policyholders och group of policyholders i.If X_i denotes the total claim cost for the customer and w_i in this case the number of claims for the corresponding customer then $Y_i = X_i/w_i$, the quantity of interest, is defined as average claim cost. Note that w_i is in this case considered deterministic and observed at the time of the analysis. In addition, an important assumption in the standard setting is that number of claims and claims costs are independent. As with the other assumptions, the assumption of independent claim counts and

can be relaxed as shown in [10]. Choosing the distribution of X depends on a number of preferences. Arguing a distribution which is on the positive real line is not difficult since the insurance company does not reimburse negative claim costs as a standard. Another desirable property is that the distribution has a nice form under summation and scaling of the probability distribution. There are many distributions that satisfy these properties but the Gamma distribution has become an industry standard for insurance applications [21]. With $w_i = 1$ the gamma distribution is

$$f(x_i) = \frac{\beta^\alpha}{\Gamma(\alpha)} x_i^{\alpha-1} e^{-\beta x}, \qquad x > 0 \tag{33}$$

with $\alpha > 0$ and $\beta > 0$. Applying the transformation $Y_i = X_i/w_i$ transforms the distribution in (33) to

$$f_{Y\,i}(y_i) = w_i f_{X\,i}(w_i y_i) = \frac{(w_i \beta^{w_i \alpha})}{\Gamma(w_i \alpha)} y_i^{w_i - 1} e^{-w_i \beta y_i}. \tag{34}$$

The Gamma distribution is also part of the EDM but it will not be shown here.[22]

2.3.3 Maximum Likelihood Estimation for GLM

As explained i earlier section maximum likelihood estimation gives a parameter estimate which fits a distribution to the observed values of $(Y_1, ..., Y_n)$ which are most probable given the assumed distribution they belong to. The output from the estimation is a set of parameters for the distribution.

For the EDM family with the observed values $(Y_1, Y_2, ..., Y_n)$ the likelihood function is

$$L(\theta) = \prod_{i=1}^{n} \exp\left(\frac{Y_i\theta - b(\theta)}{\phi/w_i} + c(Y_i, \phi, w_i)\right) \tag{35}$$

where $(Y_1, ..., Y_n)$ is the observation of interest and w_i is the exposure value. Expression (35) is estimated with respect to θ. Maximizing (35) is equivalent to maximizing the log of (35)

$$\log L(\theta) = \sum_{i=1}^{n} \left(\frac{Y_i\theta - b(\theta)}{\phi/w_i} + \log c(Y_i, \phi, w_i)\right) \tag{36}$$

giving a easier function to work with. By defining that $\theta_i = g^{-1}(\mathbf{X_i}^T \boldsymbol{\beta})$ where $\boldsymbol{X_i} = (X_1, X_2, ..., X_p)$ are the observed explanatory parameters of the observations and $\boldsymbol{\beta}$ are the corresponding parameters for the GLM which need to be estimated. Maximizing (36) with respect to $\boldsymbol{\beta}$ is procedurally similar to maximizing any type of function. Taking the derivative of (36) with respect to a β

gives then

$$\begin{aligned}\frac{\partial \log L(\theta)}{\partial \beta_j} &= \sum_{i=1}^{n} \frac{\partial}{\partial \beta_j}\left(\frac{Y_i\theta_i - b(\theta_i)}{\phi/w_i}\right) \\ &= \sum_{i=1}^{n} \frac{\partial}{\partial \theta_i}\left(\frac{Y_i\theta_i - b(\theta_i)}{\phi/w_i}\right)\frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \beta_j} \end{aligned} \tag{37}$$

with $\mu_i = b'(\theta_i)$ and in situations where $g()$ is a canonical link it holds that

$$g\big(b'(\theta_i)\big) = \theta_i. \tag{38}$$

The maximization in the procedure requires solving

$$\sum_{i=1}^{n} \frac{\partial}{\partial \theta_i}\left(\frac{Y_i\theta_i - b(\theta_i)}{\phi/w_i}\right)\frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \beta_j} = 0. \tag{39}$$

With the addition that $\partial\mu_i/\partial\theta_i = b''(\theta_i)$,the variance of μ_i being $v(\mu_i) = b''(\theta_i)$ and finally that $\partial g(\mu_i)/\partial\beta_i = X_{ij}$ simplifies the problem to

$$\sum_{i=1}^{n} \frac{Y_i - \mu_i}{v(\mu_i)g'(\mu_i)}X_{ij} = 0. \tag{40}$$

See [22] for proofs.

2.4 Generalized Additive Models in Non-Life Insurance Pricing

One drawback of applying GLM in insurance pricing is the handling of continuous variables which are used as rating factor for the pricing of the insurance policies. For example these would include the policyholders age or residence distance form a highly busy highway. Using these variables in the standard GLM setting usually resolves in having to group the variables into distinct group and use dummy variables in the estimation algorithms. The dummy variable method fits well when there are naturally distinct groups that fits well. However there are more naturally fitting models to parameters which can behave continuously.

The application of continuous estimation of explanatory variables in the general setting is done using smoothing splines to estimate the risk based on a continuous value. Splines are constructed to form a linear function from other piece wise continuous functions to fit some smooth line [15]. The additive model can be, in its simplest form, expressed as

$$\eta_i = \beta_0 + f_1(x_{i1}) + ... + f_J(x_{iJ}) \tag{41}$$

for observation of policyholder or group of policyholders i for some general functions f_j which in the application of this thesis is splines which are not more

complex than being polynomials. One example of such functions is the cubic spline on the form

$$p_i(t) = a_i + b_i(t - t_i) + c_i(t - t_i)^2 + d_i(t - t_i)^3 \tag{42}$$

where there are points $t_1, ..., t_m$ referred to as knots where on a closed sub interval $[t_i, t_{i+1}]$. Expression (42) has favorable properties such as being twice differentiable. The estimation problem essentially reduces to minimizing a cost function where a part of that function are defined as cubic splines. In numerical applications there are a few alternatives to parameterize and apply cubic splines numerically. One such solution is the application of B-splines which from a linear combination form a set of basis for the cubic splines [7]. The definition of such basis splines is for the zeroth spline

$$\begin{aligned} B_{0,i}(t) &= \begin{cases} 1, & t \in [t_i, t_{i+1}) \\ 0, & \text{otherwise} \end{cases} \qquad i = 1, ..., m-2 \\ B_{0,i}(t) &= \begin{cases} 1, & t \in [t_i, t_{i+1}] \\ 0, & \text{otherwise} \end{cases} \qquad i = m-1 \end{aligned} \tag{43}$$

where m is the number of points through which the smoothing function is to be smoothed. To estimate the rest of the splines the recursion

$$\begin{aligned} B_{k+1,1}(t) &= \frac{t_2 - t}{t_2 - t} B_{k,1}(t) \\ B_{k+1,i}(t) &= \frac{t - t_{max(i-k-1,1)}}{t_{min(i,m)} - t_{max(i-k-1,1)}} B_{k,i-1}(t) + \frac{t_{min(i+1,m)} - t}{t_{min(i+1,m)} - t_{max(i-k,1)}} B_{k,i}(t) \\ B_{k+1,m+k}(t) &= \frac{t - t_{max(m-1,1)}}{t_m - t_{m-1}} B_{k,m+k-1}(t) \end{aligned} \tag{44}$$

for $i = 2, ..., m + k - 1$. When $k = 2$ then (45) is a cubic spline. The spline can be, in term of basis functions, summarized as

$$s(t) = \sum_{i=1}^{m+2} \beta_i B_{3,i}(t) \tag{45}$$

where $\beta_1, ..., \beta_{m+2}$ are parameters to be estimated.

2.4.1 Poisson Case

To estimate the parameters of the GAM in the Poisson case a deviance function defined as in (23). For the Poisson case without continuous variables the deviance becomes as in (20). Introducing a continuous variable in estimating equation requires an introduction of penalty term in (20) the expression to minimize becomes

$$\Delta(f) = D(\boldsymbol{Y}, \hat{\mu}) + \lambda \int_a^b (f''(x))^2 dx \tag{46}$$

with a penalty term λ. More specifically (46) becomes, with a single continuous variable,

$$\Delta(s) = 2\sum_{i=1}^{n} = w_i(Y_i \log(Y_i) - Y_i s(X_i) - Y_i + \exp\{s(X_i)\}) + \lambda \int_a^b (f''(x))^2 dx \tag{47}$$

called the penalized deviance. In (47) $\exp(s(X_i)) = \mu_i$ where $s(X)$ is the spline. Writing (47) in terms of the expressions in this chapter (β from (45)), (47) becomes for a set of parameters $\boldsymbol{\beta} = (\beta_1, ..., \beta_{m+2})$

$$\begin{aligned}\Delta(\boldsymbol{\beta}) = 2\sum_{i=1}^{n} w_i \left(Y_{ii} - Y_i \sum_{j=1}^{m+2} \beta_j B_j(x_i) - Y_i + \exp\left\{ \sum_{j=1}^{m+2} \beta_j B_j(x_i) \right\} \right) \\ + \lambda \sum_{j=1}^{m+2} \sum_{k=1}^{m+2} \beta_j \beta_k \Omega_{jk}.\end{aligned} \tag{48}$$

Ω_{jk} in (48) is defined as

$$\int_{z_1}^{z_m} B_j''(x) B_k''(x) dx \tag{49}$$

with the properties that

$$B'_{j+1,k} \frac{j+1}{u_k - u_{k-j-1}} B_{j,k-1}(x) - \frac{j+1}{u_{k+1} - u_{k-j}} B_{j,k}(x), \tag{50}$$

$$B'_{j+1,k} \frac{j+1}{u_k - u_{k-j-1}} B_{j,k-1}(x) - \frac{j+1}{u_{k+1} - u_{k-j}} B_{j,k}(x), \tag{51}$$

$$B''_{j+1,k} \frac{j+1}{u_k - u_{k-j-1}} B'_{j,k-1}(x) - \frac{j+1}{u_{k+1} - u_{k-j}} B'_{j,k}(x), \tag{52}$$

for a set of knots (points where the piece wise splines meet) $u_1, ..., u_m$. Expression (48) is then solved for the vector of parameters $\boldsymbol{\beta}$ by applying some numerical method such as Newton-Rhapson. [22]

2.4.2 Gamma Case

For the Gamma distribution case, the same approach is applied but with the deviance

$$D(Y, \lambda) = 2\sum_{i=1}^{n} w_i(Y_i/\lambda_i - 1 - \log(Y_i/\lambda_i)). \tag{53}$$

For the single variable case, (53) is then as in the Poisson case changed such that $\lambda_i = \exp(s(X_i))$ where again $s(X)$ is the spline. Expression (53) is then solved for its vector $\boldsymbol{\beta}$ to find the parameters for the model. [22]

2.5 Tree Based Methods

Moving beyond the classical regression methods gives rise to a number of methods with similar objectives but different methodologies where the models are not restricted for the same effects for all other parameters unless specified. Regression trees build on the more classification oriented methods of decision trees where, as in the regression models, the goal is to form conclusions about a target variable given the observation of the input variables.[3]

2.5.1 Regression Tree

The main difference between decision trees and regression trees is that decision trees are mainly use to predict the outcome of a qualitative response while regression trees is used to predict the outcome of a quantitative response. Comparing with the classical regression models on the form

$$f(X) = \beta_0 + \sum_{i=1}^{p} X_i \beta_i, \tag{54}$$

regression trees are similar but is instead

$$f(X) = \sum_{i=1}^{n} c_i \mathbf{1}_{X \in R_i} \tag{55}$$

with $(R_1, ..., R_n)$ representing a partition of the feature space. The feature space is the space mapped from the explanatory variables $(X_1, ..., X_p)$ to the predictions denoted by R. For two variables X_1 and X_2 the partition can look like in Figure 1.

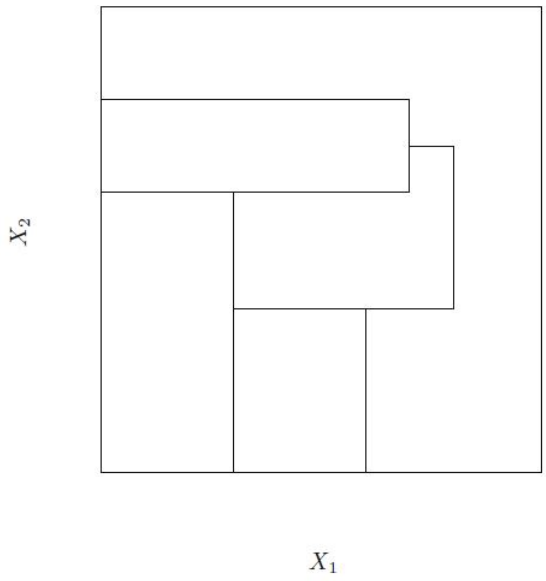

Figure 1: Partition for two explanatory variables X_1 and X_2.

Considering Figure 1, the partition maps from two variables to five outcomes. Estimating procedure for regression trees starts from the set of $\boldsymbol{X} = (X_1, ..., X_p)$ predictor variables and the response variable Y. Observing the pairs $(\boldsymbol{X}_i, Y_i)$ for observation of policyholder i assuming they are i.i.d. the partition is then done into a set of regions $(R_1, ..., R_n)$ using recursive binary splitting. The estimation starts from the full predictor space with the start of splitting variable j and a split-point $s \in \mathbb{R}$ resulting in that the predictor space $R = R_1(j, s) \cup R_2(, s)$. In this first instance R is then

$$\begin{aligned} R_1(j, s) &= \{\boldsymbol{x} \in \mathbb{R}^p : x_j \leq s\} = \mathbb{R} \times \mathbb{R} \times, ..., \times (-\infty, s] \times \mathbb{R} \times, ..., \times \mathbb{R} \\ R_2(j, s) &= \{\boldsymbol{x} \in \mathbb{R}^p : x_j > s\} = \mathbb{R} \times \mathbb{R} \times, ..., \times (s, \infty) \times \mathbb{R} \times, ..., \times \mathbb{R}. \end{aligned} \tag{56}$$

From (56), the algorithm continues by splitting one of the regions into two more regions and so on until a stopping rule criterion is reached. The goal is to find an estimator $\hat{c}_n$ for c_n for (55).

As mentioned in the statistical learning section, the general principle for estimating a regression tree is to define some kind of evaluation of how close the actual prediction is to the real value. As in the rest of this thesis and as specified in (26), the measure of fit is chosen to be the Poisson deviance for the claim frequency in algorithm 1 [3] adjusted for the Poisson distribution.[16]

Algorithm 1 CART for Poisson Predicted Variable

1.Initialization:

Initialize the split

$$R_1(j,s) = \{\boldsymbol{x} : x_j \leq s\}$$
$$R_2(j,s) = \{\boldsymbol{x} : x_j > s\},$$

and find j and s by finding the solution to

$$\min_{j,s}\left[\min_{\mu_1}\sum_{\{i:x_i\in R_1(j,s)\}} 2\left(w_iY_i\log\left(\frac{Y_i}{\mu_1}\right) - w_i(Y_i-\mu_1)\right)\right.$$
$$\left.+\min_{\mu_2}\sum_{\{i:x_i\in R_2(j,s)\}} 2\left(w_iY_i\log\left(\frac{Y_i}{\mu_2}\right) - w_i(Y_i-\mu_2)\right)\right].$$

μ_1 and μ_2 are then estimated by

$$\hat{\mu}_1 = \frac{1}{\sum\limits_{\{i:x_i\in R_1(j,s)\}} w_i}\sum_{\{i:x_i\in R_1(j,s)\}} N_i$$

and

$$\hat{\mu}_2 = \frac{1}{\sum\limits_{\{i:x_i\in R_2(j,s)\}} w_i}\sum_{\{i:x_i\in R_2(j,s)\}} N_i$$

2.Recursion:

Repeat step one for the rest of the regions to partition R into $|R|$ potential regions. The new region is chosen such that the deviance is minimized over the whole region. Resulting in

$$K^+ = \arg\min_{k\in\{1,\ldots,|R|\}}\sum_{i=1}^{n} 2\left(w_iY_i\log\left(\frac{Y_i}{\hat{\mu}_n^k}\right) - w_i(Y_i-\hat{\mu}_n^k)\right)$$

with k being the k:th partition of the region R.

3.Repeat:

Repeat 2 until a stopping rule is fulfilled.

Before choosing the final tree in the estimation procedure and the number of partitions, there has to be rules defined for how to choose the number of leaves in the tree. Compared to other models and methods to predict outcomes from a specific input, regression trees need also to handle the problem of over fitting. Specifying a regression tree which is fully partitioning data results in a tree for which every data point in the data set. Therefore a regression tree need to be less complex than having a perfect fit to the data. Starting with a perfect fit and then excluding splits backwards systematically is a strategy which might

be feasible to find a tree with a acceptable fit to some measure. In essence, the strategy will result in a sub tree from the original very large tree. Iteratively this strategy can be formulated as specifying a sub tree, running a cross validation on the predictions of the sub tree and saving the sub tree if it has a lower prediction error than other sub trees. The strategy however will result in many sub trees which will then be computationally heavy to find the best tree among these. Another more efficient strategy is necessary to find a adequate tree or to limit the set of sub trees. This strategy is generally referred to cost complexity pruning and applies an additional parameter which is used to index a sequence of trees[16]. The estimation algorithm with the α parameter results in a adjusted loss function to

$$\sum_{n=1}^{|T|} \sum_{\{i:x_i \in R_n\}} 2\left(w_i Y_i \log\left(\frac{Y_i}{\hat{\mu}_n}\right) - w_i(Y_i - \hat{\mu}_n)\right) + \alpha|T| \tag{57}$$

with $|T|$ being the size of the tree and α a tuning parameter which needs to be specified. Expression (57) is designed such that large trees get penalized compared to small trees. n is the number of regions in (57) for the trees. If n is as large as the number of observations and α is small then the tree would be the maximum of the size. On the contrary having a α which is increasing all the trees will have a penalty term which is large and the function will be the observed average.

Summarizing the cost complexity pruning as an algorithm gives an estimate to α and to the tree. An algorithm for finding the tree size as well as the cost parameter can be specified as

Algorithm 2 CART for Poisson Predicted Variable

1.Initialization:
Initialize the regression tree by running the splitting algorithm

2.Recursion:
Apply cost complexity pruning to find the set of best sub trees as a function of α as described in the next step.

3.-fold cross validation:
Apply K-fold cross validation to find α such that steps 1 and 2 are repeated on the training data from the specification of the K-fold cross validation and find the prediction error as a function of α and finally pick α such that the prediction error is minimized.

2.5.2 Random Forest

Random forests are an extension of the regression tree as hinted in the name. Regressions trees are suitable for the training set but have poor performance on a test set [16]. A natural property of trees which are very deep (have a large

number of partition) is that they will perfectly fit the data and cause over fitting issues. By building the predictions using multiple trees instead of one the errors or over fitting issues get averaged out.

Aggregating a prediction works in the sense that a set of p observed response variables and explanatory variables $\mathcal{P} = \{(\boldsymbol{X}_1, Y_1), (\boldsymbol{X}_2, Y_2), ..., (\boldsymbol{X}_p, Y_p)\}$. Applying some prediction algorithm a predictor $\hat{f}(\boldsymbol{X}, \mathcal{P})$ of the true function $f()$ can be obtained as explained in the previous sections. If there instead exists or is constructed as sequence of sets of data $\{\mathcal{P}_k\}_{k\geq 1}$ the aggregated predictor can be for example the average of the predictors of this set defined as

$$\hat{f}_{agg}(\boldsymbol{X}) = \frac{1}{K}\sum_{k=1}^{K} \hat{f}(\boldsymbol{X}, \mathcal{P}_k). \tag{58}$$

Expression (58) works well because of the property of law of large numbers [16] that

$$\lim_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} \hat{f}(\boldsymbol{X}, \mathcal{P}_k) \xrightarrow{P} E[\hat{f}(\boldsymbol{X}, \mathcal{P})]. \tag{59}$$

The details and proof of why aggregation does not change bias but increases predictability will not be discussed here but the reader can refer to [16].

If observed data is used instead of simulated, it is difficult to obtain a sufficiently large enough number of samples to estimate $f()$. Bootstrapping is therefore applied to find the aggregated estimator for $f()$. Usually the methodology is performed by re-sampling with replacement a sequence of samples of $\{\mathcal{P}\}$ such that from $\{\mathcal{P}^{(b)}\}_{b=1,...,B}$ a aggregated bootstrap predictor

$$\hat{f}_{bagg}(\boldsymbol{X}) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}(\boldsymbol{X}, \mathcal{P}^{(b)}). \tag{60}$$

Ideally bagging estimators will reduce the variance of the predictors. Random forests are an extension of the bagging procedure with the tweak that samples will be de correlated. At each re sampling of the data set, the algorithm instead chooses a subset of the predictor variables $\sqrt{p}$ instead of p. The algorithm is outlined in algorithm table 2.[16]

Algorithm 3 Random Forest

Consider the B as the number of sub samples from observations of $\mathcal{P}$
for i in $1, ..., B$ **do**
Generate a sample from B^* from B
Select m variables randomly from p possible variables.
Using the m variables. Split the tree according the the previously defined algorithm. For example CART.
The result is a tree from the random forest $\hat{f}(\boldsymbol{X}, \boldsymbol{\mathcal{L}}(\mathcal{P}))$
end for
Calculate

$$\hat{f}_{rf}^{B}(\boldsymbol{X}) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}(\boldsymbol{X}, \mathcal{L}_b(\mathcal{P})).$$

Algorithm 2 in essence estimates many regression trees on different samples of the underlying data and calculates the average of all predictors to find an estimate.

2.6 Neural Networks

Neural networks or artificial neural networks have recently gained popularity as a result of increasing computing power, development of algorithms, better hardware and increasing data collection capabilities. Recent areas of development and applications of neural networks include image classification and speech recognition for example. The idea behind neural networks were originally to mimic the neurons of the brain and use them for prediction or classification purposes.[12] Mathematically a neural network is essentially a non-linear function describing a output Y in terms of its observed characteristics $\boldsymbol{X} = (X_1, X_2, ..., X_p)$ weighted with the parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_p)$ such that

$$Y = f(X_1, X_2, ..., X_p; \theta) + \epsilon. \tag{61}$$

The error term ϵ of the neural network estimation will not be presented in the estimated quantity of Y, $\hat{Y}$ meaning that the estimate will try to capture the systematical components rather than the noise. To introduce non-linearity to (61) a scalar function referred to as the activation function, $\sigma : \mathbb{R} \to \mathbb{R}$, is applied to $\boldsymbol{X} = (X_1, X_2, ..., X_p)$ instead of $f()$ in (61) similarly as in the GLM application described in previous sections. Choosing which activation function to apply is not trivial and there are many choices in the literature such as the rectified linear unit (ReLU)[12]

$$\sigma(x) = \max(0, x). \tag{62}$$

In this thesis the sigmoid function

$$\sigma(x) = \frac{1}{1 + e^{-\delta x}} \tag{63}$$

is applied to the layers with the parameter $\delta = 1$ since the objective is to estimate a positive quantity which will also be between 0 and 1 and in the estimation procedure.[12] In many senses a neural network on this form is very much a generalized linear model. Building the neural network has however a key difference which can be described as adding more layers with additional activation functions between the input and the output. Defining a hidden unit in the neural network such as

$$H_i = \sigma(\beta_{0i} + \beta_{1i}X_1 + \beta_{2i}X_2 + ... + \beta_{pi}X_p) \tag{64}$$

where i is the i-th hidden unit in the network is essentially setting parameters as weights for the inputs and running through a activation functions not much different from a GLM. Figure 2 illustrates a neural network with one hidden layer and output as an estimate.

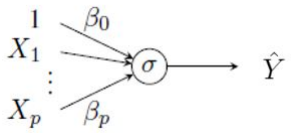

Figure 2: A neural network with one hidden layer.

Proceeding until some criteria is met, there can be more hidden units defined as

$$\begin{aligned} H_1 &= \sigma(\beta_{01}^{(1)} + \beta_{11}^{(1)}X_1 + \beta_{21}^{(1)}X_2 + ... + \beta_{p1}^{(1)}X_p) \\ H_2 &= \sigma(\beta_{02}^{(1)} + \beta_{12}^{(1)}X_1 + \beta_{22}^{(1)}X_2 + ... + \beta_{p2}^{(1)}X_p) \\ &\vdots \\ H_M &= \sigma(\beta_{0M}^{(1)} + \beta_{1M}^{(1)}X_i + \beta_{2M}^{(1)}X_2 + ... + \beta_{pM}^{(1)}X_p). \end{aligned} \tag{65}$$

Having two layers results in that the output $\hat{Y}$ is defined by the hidden units as

$$\hat{Y} = \sigma(\beta_0^{(2)} + \beta_1^{(2)}H_1 + \beta_2^{(2)}H_2 + ... + \beta_M^{(2)}H_M) \tag{66}$$

with p explanatory variables and M hidden units having the structure as in Figure 3.

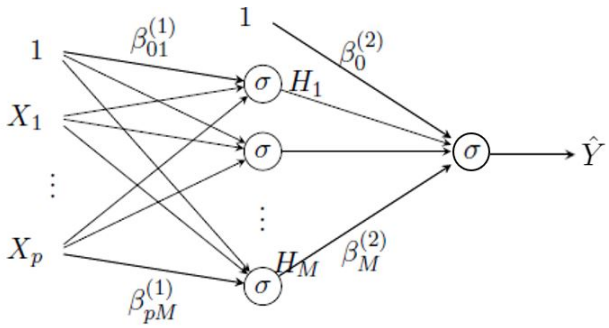

Figure 3: A neural network with M hidden units.

With the same approach as developing a a network with two layers, extending the network to multiple layers as in (64) results in a multi layer neural network. Similarly to Figure 3 extending the network is simply adding more hidden layers where the output form the first layer is input into the second and so on.
Rewriting (70) in matrix notation with $\boldsymbol{\beta}^{(1)} = [\beta_{01}^{(1)}, \beta_{02}^{(1)} ... \beta_{0M}^{(1)}]$,

$$\mathbf{W}^{(1)} = \begin{bmatrix} \beta_{11}^{(1)} & \cdots & \beta_{1M}^{(1)} \\ \vdots & \ddots & \vdots \\ \beta_{p1}^{(1)} & \cdots & \beta_{pM}^{(1)} \end{bmatrix},$$

$\boldsymbol{\beta}^{(2)} = [\beta_0^{(2)}]$ and $\boldsymbol{W}^{(2)} = [\beta_1^{(2)} ... \beta_M^{(2)}]^{\mathrm{T}}$ giving (64) and (70) in matrix notation

$$\boldsymbol{H} = \sigma_1(\boldsymbol{W}^{(1)T}\boldsymbol{X} + \boldsymbol{\beta}^{(1)T}) \tag{67}$$
$$\hat{\boldsymbol{Y}} = \sigma_2(\boldsymbol{W}^{(2)T}\boldsymbol{H} + \boldsymbol{\beta}^{(2)}) \tag{68}$$

where $\sigma_1()$ and $\sigma_2()$ does not have to be the same function. Extending (69) and (68) to include more layers denoted l is a exercise of stacking each layer l to include the input from the previous layer $(l-1)$ such that

$$\boldsymbol{H}^{(l)} = \sigma_l(\boldsymbol{W}^{(l)T}\boldsymbol{H}^{(l-1)} + \boldsymbol{\beta}^{(l)T}). \tag{69}$$

and that each layer $\boldsymbol{H}^{(l)} = [H_1^{(l)}, ..., H_{Ml}^{(l)}]$ consist of a number of hidden units which can be of different dimensions. Summarizing the layers results in

$$\begin{aligned} \boldsymbol{H}^{(1)} &= \sigma(\boldsymbol{W}^{(1)T}\boldsymbol{X} + \boldsymbol{\beta}^{(1)T}) \\ \boldsymbol{H}^{(2)} &= \sigma(\boldsymbol{W}^{(2)T}\boldsymbol{X} + \boldsymbol{\beta}^{(2)T}) \\ &\vdots \\ \boldsymbol{H}^{(L-1)} &= \sigma(\boldsymbol{W}^{(L-1)T}\boldsymbol{H}^{(L-2)} + \boldsymbol{\beta}^{(L-1)T}) \\ \hat{Y} &= \sigma(\boldsymbol{W}^{(L)T}\boldsymbol{H}^{(L-1)} + \boldsymbol{\beta}^{(L)T}) \end{aligned} \tag{70}$$

which is a neural network with L layers. Collecting the parameters and denoting the vector to estimate results in

$$\boldsymbol{\theta} = [vec(\boldsymbol{W}^{(1)})^T \;\; vec(\boldsymbol{W}^{(2)})^T \ldots \;\; vec(\boldsymbol{W}^{(L)})^T \;\; \boldsymbol{\beta}^{(1)T} \;\; \boldsymbol{\beta}^{(2)T} \ldots \boldsymbol{\beta}^{(L)T}]^T$$

(71)

solved for with some kind of minimization of error or maximization of likelihood. The matrix $\boldsymbol{W}^{(1)}$ has dimensions $p \times M_1$ and $\boldsymbol{\beta^{(1)}}$ has dimension $1 \times M_1$ and so forth.

2.6.1 Estimation of Parameters With Back Propagation

Estimating the vector of parameters $\boldsymbol{\theta}$ is similar to the estimation of parameters in a general setting by solving

$$\hat{\boldsymbol{\theta}} = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} L(X_i, Y_i, \theta) \tag{72}$$

by setting a predefined loss or deviance function L. By having the objective function as the Poisson deviance from (26), the analysis data is adjusted to estimate the parameters for predicting a claim frequency in the insurance context. Applying a neural network which takes an input through the network and gives an output without circling back the result from one layer to a previous layer is refereed to as a feed forward neural network. To efficiently estimate the parameters of the feed forward neural network a algorithm called back propagation is deployed.[13] As the objective of the estimation is to minimize a function of loss or deviance, the estimations usually will involve the computation of some derivatives. In addition, as it is likely that the estimation problem will have a multi dimensional loss function, there will be a need for the computation of the gradient with respect to each parameter. Naturally a naive computation of the gradient with respect to each weight will be computationally difficult. An alternative approach is to deploy the chain rule starting with computing the gradient at the loss function and the iterating backwards by then computing the gradient at each layer it the network. Such an algorithm is called back propagation and the approach avoids redundant calculation of the intermediate terms in the chain rule.

Illustrating the back propagation algorithm with $\boldsymbol{X}$ as input of the observed features, Y as target output, L as the loss function or deviance function, $\boldsymbol{W}^{(l)}$ as the weight matrix at layer l and σ as the activation function at each layer. Expressing the output of neural network in a loss or error function in terms of the matrix weights and the input matrix iteratively as

$$\hat{\boldsymbol{Y}} = \sigma^{(L)}(\boldsymbol{W}^{(L)}\sigma^{(L-1)}(\boldsymbol{W}^{(L-1)}...\sigma^{(1)}(\boldsymbol{W}\boldsymbol{X})...)) + \boldsymbol{\beta}^{(L)} \tag{73}$$

gives the possibility of the iterative application of the chain rule backwards. There are a few derivatives from the neural network which need to be defined to be able the compute the gradient. Firstly the derivative of the loss with respect to the weight at the layer l is

$$\frac{\partial L}{\partial \boldsymbol{W}^{(l)}} = \frac{\partial L}{\partial \boldsymbol{U}^{(l)}} \frac{\partial \boldsymbol{U}^{(l)}}{\partial \boldsymbol{W}^{(l)}} \tag{74}$$

by applying the chain rule and with setting a intermediate term for $\sigma^{(L)}(\boldsymbol{W}^{(L-1)})+\boldsymbol{\beta}^{(l)} = \boldsymbol{U}^{(l)}$ and with respect to the additive term as

$$\frac{\partial L}{\partial \boldsymbol{\beta}^{(l)}} = \frac{\partial L}{\partial \boldsymbol{U}^{(l)}} \frac{\partial \boldsymbol{U}^{(l)}}{\partial \boldsymbol{\beta}^{(l)}}. \tag{75}$$

Solving (75) requires the derivatives $\frac{\partial L}{\partial \boldsymbol{U}^{(L)}}, \frac{\partial L}{\partial \boldsymbol{U}^{(l)}}$, $\frac{\partial \boldsymbol{U}^{(l)}}{\partial \boldsymbol{W}^{(l)}}$ and $\frac{\boldsymbol{U}^{(l)}}{\partial \boldsymbol{\beta}^{(l)}}$ with (L) being the final layer. The derivatives needed are

$$\frac{\partial L}{\partial \boldsymbol{U}^{(L)}} = \frac{\partial L}{\partial \sigma(\boldsymbol{U}^{(L)})} \frac{\partial \sigma(\boldsymbol{U}^{(L)})}{\partial \boldsymbol{U}^{(L)}} = \frac{\partial L}{\partial \sigma(\boldsymbol{U}^{(L)})} \circ \sigma'(\boldsymbol{U}^{(L)}) \tag{76}$$

where $\circ$ is the element wise multiplication of the matrix,

$$\sigma'(\boldsymbol{U}^{(L)}) = \sigma(\boldsymbol{U}^{(L)})(1 - \sigma(\boldsymbol{U}^{(L)})), \tag{77}$$

$$\frac{\partial L}{\partial \boldsymbol{U}^{(l)}} = \frac{\partial L}{\partial \boldsymbol{U}^{(l+1)}} \frac{\partial \boldsymbol{U}^{(l+1)}}{\partial \sigma(\boldsymbol{U}^{(l)})} \frac{\partial \sigma(\boldsymbol{U}^{(l)})}{\partial \boldsymbol{U}^{(l)}}. \tag{78}$$

Since

$$\boldsymbol{U}^{(l+1)} = \boldsymbol{W}^{(l+1)} \sigma(\boldsymbol{U}^{(l)}) + \boldsymbol{\beta}^{(l+1)} \tag{79}$$

the derivative becomes

$$\frac{\partial \boldsymbol{U}^{(l+1)}}{\partial \sigma(\boldsymbol{U}^{(l)})} = \frac{\partial}{\partial \sigma(\boldsymbol{U}^{(l)})} \boldsymbol{W}^{(l+1)} \sigma(\boldsymbol{U}^{(l)}) + \boldsymbol{\beta}^{(l+1)} = \boldsymbol{W}^{(l+1)} \tag{80}$$

and

$$\frac{\partial \sigma(\boldsymbol{U}^{(l)})}{\partial \boldsymbol{U}^{(l)}} = \sigma'(\boldsymbol{U}^{(l)}) \tag{81}$$

resulting in that

$$\frac{\partial L}{\partial \boldsymbol{U}^{(l)}} = \left(\boldsymbol{W}^{(l+1)} \frac{\partial L}{\partial \boldsymbol{U}^{(l+1)}} \right) \circ \sigma'(\boldsymbol{U}^{(l)}). \tag{82}$$

Proceeding with

$$\begin{aligned} \boldsymbol{U}^{(l)} &= \boldsymbol{W}^{(l)} \sigma(\boldsymbol{U}^{(l-1)}) + \boldsymbol{\beta}^{(l)} \\ \frac{\partial \boldsymbol{U}^{(l)}}{\partial \boldsymbol{W}^{(l)}} &= \frac{\partial}{\partial \boldsymbol{W}^{(l)}} (\boldsymbol{W}^{(l)} \sigma(\boldsymbol{U}^{(l-1)}) + \boldsymbol{\beta}^{(l)}) \\ \frac{\partial \boldsymbol{U}^{(l)}}{\partial \boldsymbol{W}^{(l)}} &= \sigma(\boldsymbol{U}^{(l-1)}). \end{aligned} \tag{83}$$

Further

$$\frac{\partial L}{\partial \boldsymbol{W}^{(l)}} = \frac{\partial L}{\partial \boldsymbol{U}^{(l)}}\sigma(\boldsymbol{U}^{(l-1)}), \tag{84}$$

$$\frac{\partial \boldsymbol{U}^{(l)}}{\partial \boldsymbol{\beta}^{(l)}} = \frac{\partial}{\partial \boldsymbol{\beta}^{(l)}}(\boldsymbol{W}^{(l)}\sigma(\boldsymbol{U}^{(l)}) + \boldsymbol{\beta}^{(l)}) = 1 \tag{85}$$

and finally

$$\frac{\partial L}{\partial \boldsymbol{\beta}^{(l)}} = \frac{\partial L}{\partial \boldsymbol{U}^{(l)}}. \tag{86}$$

Using the previous equations and derivatives for a two-layer hidden network as an example the expressions

$$\frac{\partial L}{\partial \boldsymbol{W}^{(1)}} = \frac{\partial L}{\partial \sigma(\boldsymbol{U}^{(3)})}\frac{\partial \sigma(\boldsymbol{U}^{(3)})}{\boldsymbol{U}^{(3)}}\frac{\partial \boldsymbol{U}^{(3)}}{\partial \sigma(\boldsymbol{U}^{(2)})}\frac{\partial \sigma(\boldsymbol{U}^{(2)})}{\partial \boldsymbol{U}^{(2)}}\frac{\partial \boldsymbol{U}^{(2)}}{\partial \sigma(\boldsymbol{U}^{(1)})}\frac{\partial \sigma(\boldsymbol{U}^{(1)})}{\partial \boldsymbol{U}^{(1)}}\frac{\partial \boldsymbol{U}^{(1)}}{\partial \boldsymbol{W}^{(1)}} \tag{87}$$

and

$$\frac{\partial L}{\partial \boldsymbol{\beta}^{(1)}} = \frac{\partial L}{\partial \sigma(\boldsymbol{U}^{(3)})}\frac{\partial \sigma(\boldsymbol{U}^{(3)})}{\boldsymbol{U}^{(3)}}\frac{\partial \boldsymbol{U}^{(3)}}{\partial \sigma(\boldsymbol{U}^{(2)})}\frac{\partial \sigma(\boldsymbol{U}^{(2)})}{\partial \boldsymbol{U}^{(2)}}\frac{\partial \boldsymbol{U}^{(2)}}{\partial \sigma(\boldsymbol{U}^{(1)})}\frac{\partial \sigma(\boldsymbol{U}^{(1)})}{\partial \boldsymbol{U}^{(1)}}\frac{\partial \boldsymbol{U}^{(1)}}{\partial \boldsymbol{\beta}^{(1)}} \tag{88}$$

which are solved using the derived derivative equations in this chapter. As illustrated the derivatives are calculated and applied starting with the output and going backwards to the input layer. Numerically the update of the current value of the parameters in the back propagation algorithm summarizes as

$$\boldsymbol{W}^{(l)} = \boldsymbol{W}^{(l)} - \alpha\frac{\partial L}{\partial \boldsymbol{W}^{(l)}} \tag{89}$$

$$\boldsymbol{\beta}^{(l)} = \boldsymbol{\beta}^{(l)} - \alpha\frac{\partial L}{\partial \boldsymbol{\beta}^{(l)}}. \tag{90}$$

α is called the learning rate or the tuning parameter and is the rate at which the network learn. In the application α is a additional parameter which needs to be taken into consideration when running the model fit. The algorithm for fitting the neural network is as follows,[12]

Algorithm 4 Back propagation Neural Network

Consider the n as the total number of observations in the test set of observations $\boldsymbol{X}$

0. Initialize vector of parameters

$$\boldsymbol{\theta}=[vec(\boldsymbol{W}^{(1)})^T \ vec(\boldsymbol{W}^{(2)})^T ... \ vec(\boldsymbol{W}^{(L)})^T \ \boldsymbol{\beta}^{(1)T} \ \boldsymbol{\beta}^{(2)T} ... \boldsymbol{\beta}^{(L)T}]^T$$

while Convergence criteria for not fulfilled **do**

for a vector $\boldsymbol{X}_i = (X_1, X_2, ..., X_p)$ in $1, ..., n$ **do**

1. Compute the output of the neural network. I.e. forward propagate step in the algorithm.
2. Compute the error for the output of the neural network
3. Compute the values of the derivatives in the backward pass

end for

4. Update the values of the weight vector using

$$\boldsymbol{W}^{(l)} = \boldsymbol{W}^{(l)} - \alpha \frac{\partial L}{\partial \boldsymbol{W}^{(l)}}$$

5. Break if the values of the weights are converged.

end while

6. Store the values in the vector for estimates.

$$\hat{\boldsymbol{\theta}}=[vec(\boldsymbol{W}^{(1)})^T \ vec(\boldsymbol{W}^{(2)})^T ... \ vec(\boldsymbol{W}^{(L)})^T \ \boldsymbol{\beta}^{(1)T} \ \boldsymbol{\beta}^{(2)T} ... \boldsymbol{\beta}^{(L)T}]^T$$

2.6.2 Variable Transformation

There is a sensitivity to the scale of the input variables in plenty of statistical models. If for example a model is estimated such that it takes input variables in thousands and estimates an output in thousands will produce different results if the data is suddenly input is changed to millions or hundreds. Apart from different inputs and outputs, the estimated weights can be sensitive as well. Non-scaled input variables in the training set can give estimates of the weights which are relatively large. Larger weights will then be sensitive to the input value in the test set and is likely to produce large error. Scaling in this thesis is called min-max feature scaling and is defined as

$$X_{\text{Scaled Input}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{91}$$

for continuous variables.

For categorical variables a different scaling is applied. Because of the multiplicative structure in the GLM case, having just the variable labels as groups

wont pose a problem. In the case when there is not a natural ordering of the categories for a neural network the labels might be interpreted by the model as some quantifiable order. One-hot encoding is then applied instead to circumvent the issue. The encoding creates as many dummy variables as there are levels for the variable. So if there are 3 levels of gender, one hot encoding creates 3 dummy variables with 1 when the gender of the observation belongs to the current level and 0 for the others.

2.7 Imbalanced Data

Insurance data is naturally imbalanced since insurance by nature is supposed to cover sudden accidental events. Specifying the insurance coverages usually results in that the insurance cover only covers events which are of low probability. It is not uncommon that the observed frequency of claims over a couple of years for the insurance company is $5\% - 10\%$ depending on the product and the type of insurance. Since frequencies are low by specification of the product, the data at hand for insurance companies will have a lot of observations from non claiming groups compared to the ones which are claiming. If analyst wants to find underlying structure, it might be difficult as observations of claims will be quite low. A frequency of 5% results in that 95% of the portfolio will not have any claims and the underlying structure of the data generating process will be difficult to find.

As explained in section 2, applying the duration solves the issue partly of imbalanced data sets as the observations with more duration will have more weight and thus influence the estimates more than the observations with low duration. Additional complexity is introduced when there is a low duration in groups which have a higher frequency or where the risk is considered to be higher than the rest of the policy. Models which fail capture the increased risk despite the lower exposure might end up smoothing out the risk in these individuals. The result of the smoothing is that insurance prices for high risk individuals are priced lower than their corresponding risk. There is therefore a risk that potential policyholders will take advantage of the incorrect pricing and purchase policies which are cheap.

Adjusting imbalanced data can be done with a number of approaches usually involving some form of sampling method. Random under sampling [17] for example randomly samples observations form the groups which are over represented in the data and removes them to balances out the observations. Over sampling on the other hand creates more observations from the smaller group in the data. The method applied in this thesis is called synthetic minority oversampling technique or SMOTE fort short. SMOTE does utilize a over-sampling technique but instead of just oversampling of the smaller group, new synthetic observations are created for the group which is under represented in the data out of the existing observations [5].

Creating the synthetic observations is done by selecting a random observation from the under represented group in the data which contains the observation variable and all of the explanatory variables. A group of new observations form the k-nearest neighbours of the first selected observation. K-nearest neighbours are then used to create a new synthetic observation from the data. In this thesis the new quantities created from the selected observation will be number of claims as well as duration. The SMOTE algorithm used in this thesis is defined as

Algorithm 5 SMOTE Algorithm

Define $D = (\boldsymbol{X}_i, Y_i)$ as the training sample for observation $i = 1, ..., n$ in the data set. Define for example extreme high cases of the response as $Y > t$ where t is for example the number of claims observed or a observed level of the frequency. Define also a relevance function such that observations above t receive relevance 1 while values below t receive relevance 0 meaning that the extreme values are more relevant in the data. This creates a subset of D, $D_r = \{(\boldsymbol{X}_i, Y_i) \in D : Y_i \geq t\}$ which are used to create synthetic observations.

0. Define D, D_r as the data set of interest and the data set considered under represented, t as the threshold defining when Y is considered extreme, $\%o$ as the percentage of oversampling of D_r i.e. how many extra samples should be created with response of interest, k as the number of nearest neighbours of each observations should be used to generate new observations.

nng $\leftarrow$ number of new observations to be generated
for all observations $\in D$ **do**
1. Set nns $\leftarrow$ kNN$(k, \text{observation}, D_r \setminus \text{observation})$
for i in $1, ...,$ nng **do**
2. Randomly choose $\boldsymbol{X}$ with replacement from nns
for all $(X_i, Y_i) \in (\boldsymbol{X}, Y)$ **do**
if X_i is numeric **then**
$X_{i \text{ diff}} \leftarrow X_{i \text{ observation}} - X_i$ {// Difference between the numeric value of the explanatory variable X_i of the observation and of the kNN X_i}
$X_{i \text{ new}} \leftarrow X_{i \text{ observation}} + \text{RANDOM}[0, 1] \times X_{i \text{ diff}}$
else
$X_{i \text{ new}} \leftarrow$ randomly select among $X_{i \text{ observation}}$ and X_i
end if
end for
3. Calculate $d_1 \leftarrow$ HEOMDistance$(X_{i \text{ new}}, X_{i \text{ observation}})$ and $d_1 \leftarrow$ HEOMDistance$(X_{i \text{ new}}, X_i)$ to find the new target value.

$$Y_{i\,new} \leftarrow \frac{d_2 \times Y_{i\,observation} + d_1 \times Y_i}{d_1 + d_2}$$

end for
end for

[28] Algorithm 5 will be applied to find the number of claims and the duration for the synthetic observations. 'HEOMdistance' is for clarification a heterogeneous distance measure where the abbreviation 'HEOM' means Heterogeneous Euclidean Overlap Metric. It is defined as

$$\text{HEOM}(\boldsymbol{X}_i, \boldsymbol{X}_j) = \sum_{r=1}^{R} d_r(X_{i,r}, X_{j,r}) \tag{92}$$

with

$$d_r(X_{i,r}, X_{j,r}) = \begin{cases} \frac{|X_{i,r} - X_{j,r}|}{\max(X_{j,r}) - \min(X_{j,r})} & \text{if } X_r \text{ is a continuous variable} \\ \delta_{i,j} & \text{if } X_r \text{ is a categorical variable} \end{cases} \tag{93}$$

for R variables with $\delta_{i,j} = 1$ if $X_{i,r} \neq X_{j,r}$ and $\delta_{i,j} = 0$ if $X_{i,r} = X_{j,r}$.[27]

2.8 Model Comparison

2.8.1 Measure of Error

Comparing and evaluating any model requires some form of measurement for how far from observed values the model prediction is. Summarizing such a measure can be done in many ways but a common approach is to use the mean of the squared error

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \left(\hat{f}(\boldsymbol{X}_i) - Y_i \right)^2. \tag{94}$$

Note that the mean squared error is not used to fit the data or update parameters but used to evaluate how well the model fits to the observed values.[8] If there exists a group of candidates of models, the aim of the comparison is to find the model which minimizes

$$E\Big[(\hat{f}(\boldsymbol{X}) - Y)^2 \Big]. \tag{95}$$

Since (95) will not be known it has to by estimated with (94). Applying the rule of total expectations to (95), it can be shown that

$$E\Big[(\hat{f}(\boldsymbol{X}) - Y)^2 \Big] = E\Big[(\hat{f}(\boldsymbol{X}) - E[Y|\boldsymbol{X}])^2 \Big] \tag{96}$$

is minimized when $\hat{f}(\boldsymbol{X}) = E[\boldsymbol{Y}|\boldsymbol{X}]$. Depending on the type of variable estimated, the evaluation function can be altered to give different estimate than the mean squared error.[8] The measurement of error for the model prediction in this thesis will be the out of sample deviance based on the log likelihood, i.e.

$$\sum_{i=1}^{n} 2(w_i Y_i \log\left(\frac{Y_i}{\hat{\mu}_i}\right) - w_i(Y_i - \hat{\mu}_i)) \tag{97}$$

with μ being the estimated frequency for each observation.

2.8.2 Cross-validation

Assessing the model performance will be done using k-fold cross validation by defining

$$CV(\hat{f}, \boldsymbol{\theta}) = \frac{1}{k} \sum_{j=1}^{k} \left(\hat{f}^{-j}(\boldsymbol{X}_j, \boldsymbol{\theta}) - \boldsymbol{Y}_j \right)^2 \tag{98}$$

as the cross validation error for j folds where $\hat{f}^{-j}(\boldsymbol{X}_j, \boldsymbol{\theta})$ is the target function estimated with parameters $\boldsymbol{\theta}$ on all folds but the j:th fold. The $j : th$ fold is used to estimate the mean square error and is called the test set. 10-fold cross validation will be applied where 10% of the data is used to test at and the rest of the data will be used to estimate the prediction error. Adjacent 10% of the data will then be used and so on.[16] As with the measure or error the cross validation will be applied with the deviance measure of error rather than the mean square error. The result for the CV will be

$$CV(\hat{f}, \boldsymbol{\mu}) = \sum_{j=1}^{k} 2(\boldsymbol{w}_j \boldsymbol{Y}_j \log \left(\frac{\boldsymbol{Y}_j}{\hat{\boldsymbol{\mu}}^{-j}} \right) - \boldsymbol{w}_j (\boldsymbol{Y}_j - \hat{\boldsymbol{\mu}}^{-j})) \tag{99}$$

with $-j$ meaning the estimate of μ is from all data but on the j:th fold. The evaluation is done by vector for all the observations in the test set j.

3 Data & Model Specification

Two data sets are used in this thesis to compare the models. One set is generated to test the models in setting where the data generating process is known and can be controlled. Set number two is real insurance data often used in insurance literature. Both data sets are commented in this section.

3.1 Simulated Data

Simulated data is generated to have specific properties for the risk in the fictional insurance portfolio. Motivating the use simulated data is the the objective of the thesis is to evaluate predictive performance of the specified models used in the insurance pricing. Knowing the data generating process behind the observations allows for the evaluation of the modelled performance compared to the true model. The generated data from the simulation has 2 continuous random variables where one has quadratic behaviour and the other a linear. Simulating number of claims requires a specification from the μ parameter for the Poisson distribution or in the case a function representing the Poisson distribution. In addition, exposures needs to be simulated to represent the proportion of insured with each risk characteristic.

The μ function used in the thesis is

$$\begin{aligned} &\mu(\text{variable}_1, \text{variable}_2) = \\ &a \times (\text{variable}_1 - b)^2 + c \times \text{variable}_2 \\ &+ d \times \boldsymbol{I}_{\{\text{variable}_1 \geq 55\} \cap \{\text{variable}_2 \geq 170\}} \end{aligned} \tag{100}$$

where the parameters a, b, c and d are chosen such that μ represents a frequency and hence becomes a value $0 < \mu \leq 1$. The threshold choices for the interaction factor in (100) are chosen arbitrarily to provide some meaningful parameter μ.

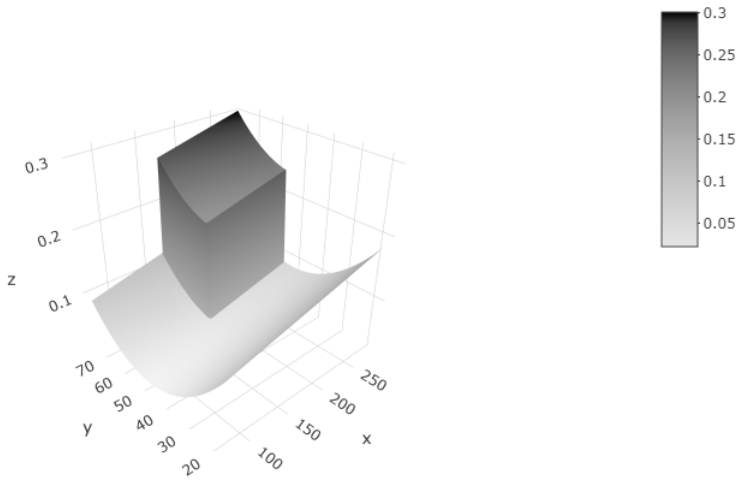

Figure 4: Surface of the frequencies derived from the μ function in (100).

The resulting shape of the frequency function is seen in Figure 4. As Figure 4 shows there is a linearly increasing z along the x-axis for y below 55. There is also an interaction effect in the upper left side of Figure 4 where z increases.

Since the objective of the thesis is to compare how the models perform in the presence of imbalanced data, the exposure is adjusted to reflect that the observed data is not uniform over all of the possible combinations of x_1 and x_2. To do so, the exposure is adjust to be less in the areas where μ changes the most. If one of the variables is assumed to represent age y in Figure 4 and variable$_1$ in (100), then the assumption is that there is considerably less exposure for low and for higher ages.

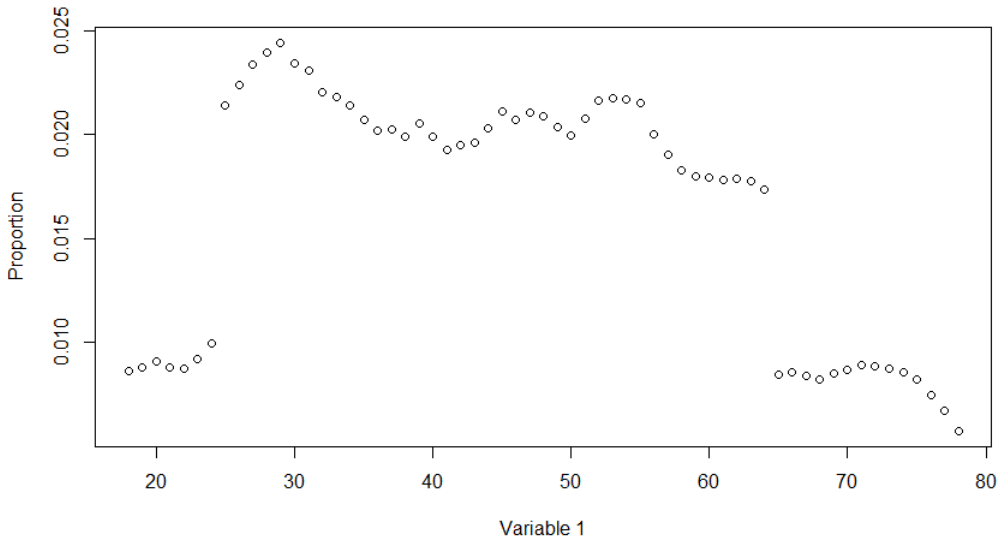

Figure 5: Proportion of the exposure by variable$_1$.

Figure 5 shows the proportion of the exposure by variable$_1$. As Figure 5 shows, the proportions in the tails are decreased to represent the imbalance in the regions. The data stems form the population distribution by age in Sweden.[4] A simulation is done for variable$_2$ such that

$$\text{variable}_2 \sim N(170, 40) \tag{101}$$

resulting in the exposure distribution in Figure 6

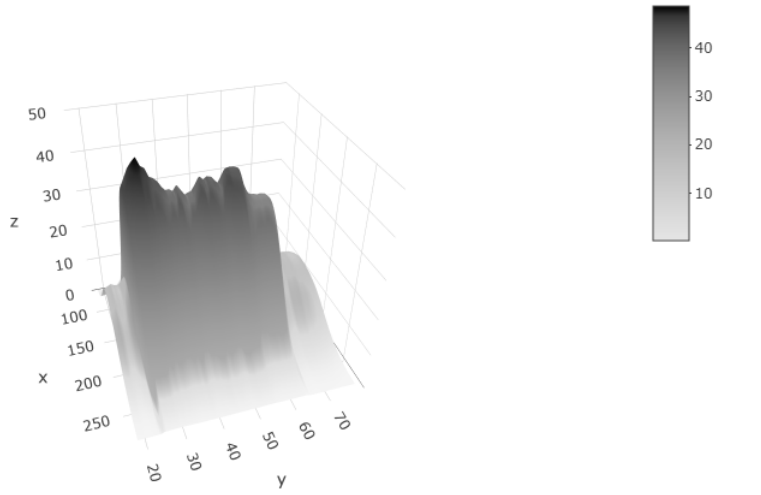

Figure 6: Total distribution of exposure in the simulated data.

The properties of the exposure distribution will then have a lot of exposure where the properties of the claims are easy to capture but a lower exposure where there are interactions and higher complexity in the claims generating process. Claims and exposure were simulated from a total number of 200 000 fictitious insurances were used in the simulated data set. This is not uncommon in practice for a large property & casualty insurance portfolio in Sweden. In summary, first data is downloaded from SCB [4] to obtain a distribution of the populations. The proportions of ages < 25 and $65 >$ are lowered such that there is clear under representation in the tails as in Figure 6. 200 000 observations are sampled form the empirical distribution of Figure 6. For each observation of the sample from variable$_1$ a second variable is sampled at random from (101) and a random uniform variable is sampled to represent the exposure during the period. Claims are then simulated from a Poisson distribution using the function from (100) based on the value of the simulated variable$_1$ and variable$_2$.

3.2 Wasa Motorcycle Insurance 1994-1998

Real data is from the former Wasa insurance company in Sweden and is from a partial casco for motorcycles during years 1994-1998.[22] A total of 64 548 observations are recorded over the period and aggregated for the policy holders with one row per observation during this period. Table 1 presents the available variables in the data set and their formats.

Variable	Explanation	Values
agarald	Owners age	integer 0-99
kon	Owners gender	M or K
zon	Geographic zone - Standard classification in Sweden	integer 1-7
mcklass	EV ratio of vehicle classification. EV = floor((Engine power × 100)/(Vehicle weight in kg))	integer 1-7
fordald	Vehicle age	integer 0-99
bonuskl	Bonus class - policy holders start at bonus class 1 and gets increased if the policy holder has a claim. If there is a claim free year the bonus class is decreased. At class 7 the holder has to have 6 consecutive claim free years to get decreased to level 6	integer 1-7
duration	Number of policy years	numeric
antskad	Number of claims	numeric
skadkost	Claim cost	numeric

Table 1: Variables and explanation of the Ohlsson motorcycle insurance data set.

The sum of the total policy insured years is 65 236 for the period and the number of claims during the same period are 697 giving a total frequency of 1.07% on the total portfolio leaving a comparably small part to estimate the characteristics which increases the risk.If a actuary estimates the risk to be 0.5% or simply just 0% she would be correct in majority of the times but the insurance company will likely loose money. Figure 8, shows the frequency and the duration by age for example and is a clear indication that the frequency is the highest where the duration is not. The majority of the exposure in year 46 and upwards does not have any claims and thus a frequency of below 2%. A question to ask is how to estimate the characteristics of the portfolio which actually have claims. Similarly Figure 8 shows the number of observations in the data set by age where it is clearly seen that the observations which have had claims are scarce.

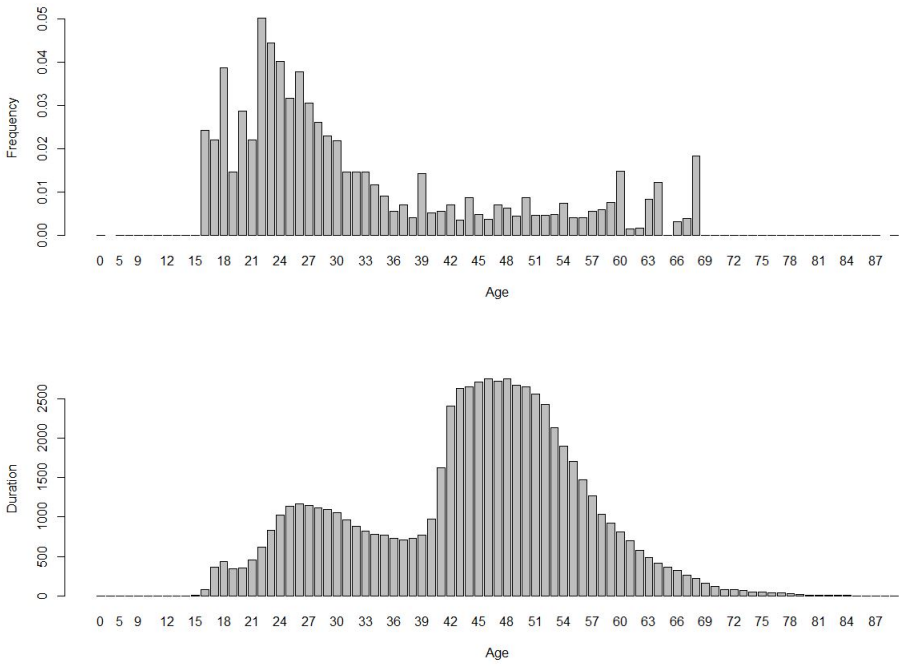

Figure 7: Frequency and exposure by age.

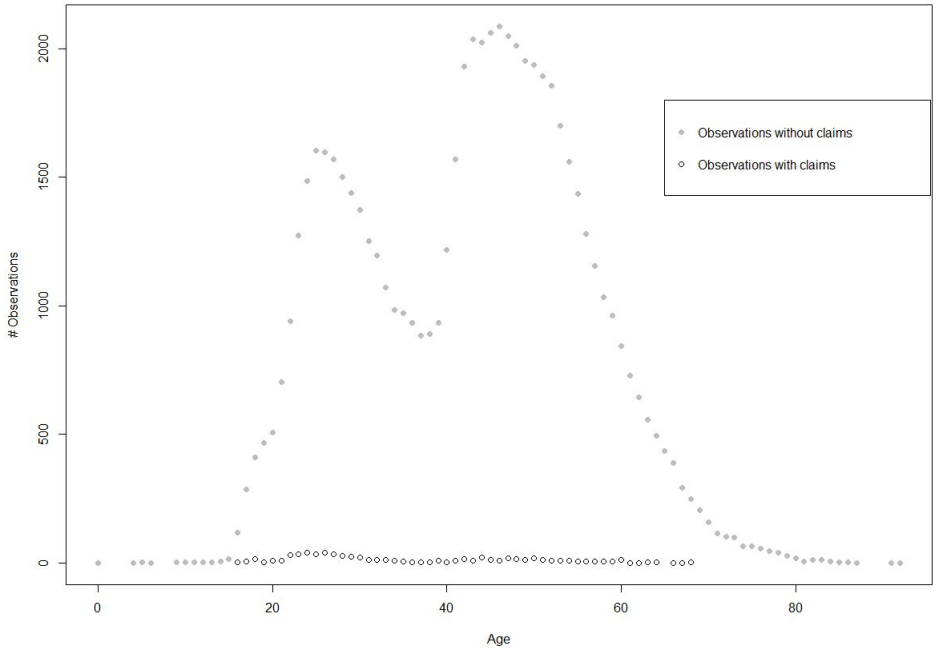

Figure 8: Frequency and exposure by age.

3.3 Model Specifications

3.3.1 GLM Specification

GLM specifications are done with various settings in the simulated data and the motorcycle insurance data. There are mainly two specifications used for the simulated data. A model without an interaction effect and a model with an interaction effect between the both variables. Continuous variables will be grouped with various settings denoted with intervals being specified in the set with lengths

$$\text{interval} = \{1, 2, 3, 4, 5\} \tag{102}$$

where the number assings the length of the interval. If 5 is applied for age for example then ages would be grouped in groups of 5. The grouping mainly concerns variables which such as age or engine power for example. Models will be specified with the Poisson setting although other choices are possible.

3.3.2 GAM Specification

In the case with simulated data GAM will be applied for both variables with and without an interaction effect. For the Wasa motorcycle data GAM will be applied for the variables which can be considered continuous and not pre

grouped, i.e. ages or powers of the vehicle. For the Wasa motorcycle data this means that there will be two variables which will be fitted with splines and and the rest as categorical variables.

3.3.3 Regression Tree and Random Forest Specification

The specifications of the regression tree and random forest is done similarly as with the GLM models. Continuous variables are grouped according to the same intervals as in the GLM case. Regression trees do not need specification of interactions for the simulated data as the model takes interactions into considerations by definition. The trees chosen as the final model are pruned according to the CV rule descried in section 2.5.1.

3.3.4 Neural Network Specification

Fitting the neural network to date causes additional problems which will affect both the estimation result as well as the estimation time. Also, since the algorithm requires starting values as input, convergence to is not guaranteed and might depend on the starting values. Choosing starting values also depends on how many parameters which should be estimated and increase with the number of layers and with the number of nodes in each layer i.e. the architecture of the neural network and there are many texts on the topic. The approach to choosing the number of input nodes and number of input layers in this thesis is to iteratively run combinations of neurons and layers and evaluate the prediction of each model. Starting with 1 hidden layer and 1 neuron and moving to a combination of 3 hidden layers and 5 neurons for the simulated data and 7 neurons for the Wasa motorcycle data. In addition and as explained earlier, the loss or deviance function will be adjusted such that duration and the Poisson distribution are taken into account.

4 Results

4.1 Hardware & Software Specifications

The specifications of the thesis gives a total 410 models to be fitted and evaluated in one of the cases. For example in the case of the motorcycle insurance data, estimating a fully connected neural network required at least 7 neurons. The specification also had two cases of 2 and 3 layers respectively resulting in 7×7 combinations of model specification in the 2 layer case and a additional $7 \times 7 \times 7$ combinations in the 3 layer case. Applying a $10-$fold cross validation to the specifications and with a total of 4 different data sets requires running the fitting algorithms approx. 16000 times where each time a model is fitted. Including each of the 100 times for the 5 specifications of the random forests for the 4 data sets and $10-$fold cross-validation gives an additional 19999 times the model needs to run on the data sets which contain at least 50000 observations. A specification of such magnitude took for example $25 - 30$ hours just to run a subset of the neural networks on a machine with 64GB RAM and a AMD Ryzen 93900X with 12 cores. As a solution a GPU-enabled Linux server with AMD Radeon Instinct MI25 with 16GB of GPU memory was used instead to run the computations. Computation time was drastically reduced to almost half compared to running the computations on a desktop machine. Good memory management is also essential since the many computations as well as the data set sets keep increasing in size while data is stored for analysis. Outcomes in the 4 cases are described in this section with the simulated data without SMOTE applied, with SMOTE applied and finally the corresponding cases from the motorcycle insurance data.

Before the CV process of section 2.8.2 was utilized, the data was randomly shuffled such to avoid any sorted columns prior to running the models. As there are a large number of models specified the reader can refer to the theoretical specifications of the models to section 3.3 for the exact models compared. The software used for this thesis was R with several different packages depending on which type of model was used. For the generalized additive models 'mgcv' in R was used without any modifications and for the generalized linear models the standard R function 'glm()' was used to fit the generalized linear regression as specified in section 3.3.1. The parameters of the fitted models where then used to predict the outcome of the test data with the function 'pred()' in R.

Own code with together with package 'rpart' was developed for implementation of the random forest to be able to apply the theory of having Poisson deviance. In each CV loop (from a total of 10 loops), a total number of 200 trees were utilized to be able to fully benefit from the method. Total number features -1 were used for each tree and were chosen at random for each tree. The trees were then used to find the prediction of the test data at each loop. Regression trees were fitted with the 'rpart' function with Poisson deviance such that the weights of observations could e taken into account.

For the neural networks, tensorflow [1] was used together with the R package for tensorflow with the GPU installation such that it utilizes the full parallel programming functionality. As mentioned in the theory section, a weighted Poisson deviance is used in this thesis and the specification required a user defined loss function with the continuity correction stated in section 2.2.1. The optimizer used from the package is 'adam' with a default learning rate of 0.0001, 100 epochs with a stopping rule which monitors the loss function implemented with the 'EarlyStopping()' function which usually triggered after approx. 60 epochs. A batch size of 10% of the data was utilized since a lot of the data could be used with the computation running on the GPU. These specifications were chosen after a few initial runs to prevent from over fitting of the whole training set and as mentioned in the beginning of this section, data was shuffled prior to defining test and validation sets. Starting values of the parameters in the neural network was just randomly sampled from a uniform distribution. The out of sample deviance of the CV was finally estimated by predicting the test data using the estimated model. Other than these specifications the specifications of the neural network architecture can be found in section 3.3.4.

To apply the SMOTE algorithm the package 'UBL' with the function 'SmoteRegress' was used with setting the relevance function such that all observations with 1 claim and above were relevant and the rest irrelevant. Distance was set to 'HEOM' (Heterogeneous Euclidean-Overlap Metric) in the function and the model was calibrated such that the observations with claims were over sampled by 20%. The algorithm for the 'SmoteRegression' is specified in section 2.7.

4.2 Simulated Data Without SMOTE

Figure 9 plots the result of the 10-fold cross validation of the models run on the simulated data from section 3 without SMOTE algorithm applied. As can be seen from the chart the GLM with both variables grouped in groups of 5 is performing well compared to the neural network models. The best performing model, i.e. the one with the lowest 10-fold CV error is the regression tree with both variables grouped in intervals of 5 although the difference on is not significantly large judging by the y-axis of the plot. The best performing neural networks are ones with 3 layers and with 5 neurons on the final layer. They do not however perform better than the GAM-model for the simulated data set without application of the SMOTE algorithm. The result show that there is not a benefit of increased complexity when the data is specified as in section 3.1 with the data having less exposure in the areas which are of interest to analyze.

Tables 2 and 3 show the in sample and out of sample deviance for each of the CV folds run in the mode. As the table shows, the GLM with grouped variables in intervals of 5 performs the second best both for in sample and out of sample data after the regression tree models. The difference in deviance is so small that it is difficult to conclude which model is better to use for prediction

purposes. It does not seem as if there is large benefit of introducing more complexity in the data. This is as expected since the data is skewed such that there is not enough exposure in the part of the portfolio where the risk has non-linear behaviour. There does not seem to be any large differences in the best performing models compared to the rest of the specifications. Comparing to Figure9 however shows that the variations of errors are not large for the rest models but models do exist were difference to the best performing model is quite large.

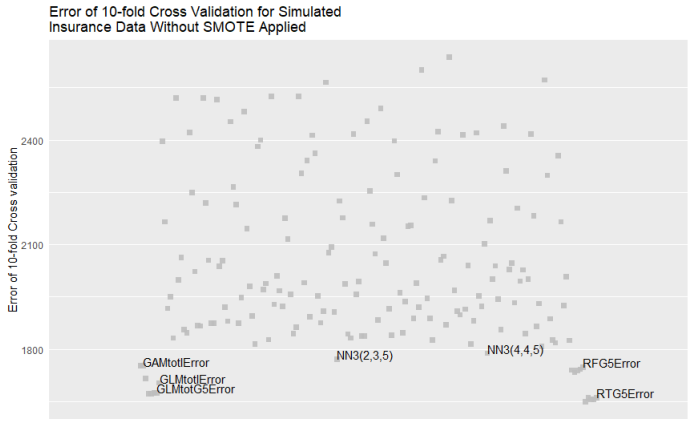

Figure 9: Cross validation errors of claims frequency for 168 different models applied on the Simulated data without SMOTE algorithm applied. "tot" indicates no grouping of continuous,"g(number)" indicated continuous variable has been grouped by interval of "number". "RF" is abbreviation for "Random Forest", "RT" is abbreviation for "Regression Tree" and "NN(n,k)" are Neural Networks with "k" hidden layers with "n" and "k" neurons in each layer.

	GLM (interaction)		**GLM (grouped variables by 5)**		**GAM**		**NN(2,3,5)**	
	In Sample	**Out of Sample**	**In Sample**	**Out of Sample**	**In Sample**	**Out of Sample**	**In Sample**	**Out of Sample**
1	15239,265	1688,563	14957,455	1657,338	15778,862	1744,098	15588,844	1736,350
2	15239,505	1688,699	14928,088	1654,190	15764,642	1770,537	15526,340	1748,472
3	15183,211	1744,712	14803,259	1701,051	15734,038	1781,430	15541,125	1818,103
4	15195,545	1732,182	14874,465	1695,581	15766,427	1776,021	15662,323	1701,062
5	15229,776	1698,021	15043,279	1677,228	15744,972	1772,383	15599,286	1752,377
6	15264,363	1663,693	14999,635	1634,839	15785,546	1727,320	15607,023	1749,708
7	15226,551	1701,286	14918,449	1666,861	15760,824	1756,946	15712,202	1660,230
8	15257,160	1670,747	15066,590	1649,878	15775,479	1736,955	15683,917	1672,650
9	15260,989	1667,109	15010,445	1639,740	15761,935	1737,108	15584,617	1782,957
10	15252,277	1675,669	14864,873	1633,108	15802,302	1728,994	15592,409	1720,147
Average	**15234,864**	**1693,068**	**14946,654**	**1660,981**	**15767,503**	**1753,179**	**15609,809**	**1734,205**

Table 2: Part 1 of the in sample and out of sample Poisson deviance of the 8 best performing models based on the 10-fold CV from out of sample predictions for the simulated data set. In sample and out of sample deviance's are presented for each of the folds from the 10-fold CV as well as the average of the CV.

	NN(4,4,5)		RF (grouped variables by 5)	
	In Sample	**Out of Sample**	**In Sample**	**Out of Sample**
1	15623,208	1740,119	15627,991	1748,850
2	15609,527	1760,349	15629,987	1746,855
3	15538,784	1817,452	15621,810	1755,032
4	15664,764	1701,370	15607,888	1768,953
5	15611,048	1753,823	15641,199	1735,643
6	15613,096	1750,117	15656,894	1719,948
7	15683,084	1657,037	15625,148	1751,694
8	15693,256	1673,747	15655,826	1721,015
9	15589,297	1783,418	15661,126	1715,716
10	15580,480	1718,796	15663,707	1713,135
Average	**15620,654**	**1735,623**	**15639,157**	**1737,684**

Table 3: Part 2 of the in sample and out of sample Poisson deviance of the 8 best performing models based on the 10-fold CV from out of sample predictions for the simulated data set. In sample and out of sample deviance's are presented for each of the folds from the 10-fold CV as well as the average of the CV.

4.3 Simulated Data With SMOTE

Figure 10 shows the 10 fold cross validated prediction error of the frequency for the simulated data set with the SMOTE algorithm applied. It is clear that the more complex models perform better in this setting in terms of predictive performance of the models. Having tree layers instead of 2 is better for the neural networks. The regression tree gives the lowest 10-fold cross validation of the errors. SMOTE models will be wrong as more samples are created which we do not know anything about other than the information from their k nearest neighbours. Given that setting GLM fails to perform as good as without any new data simulated based on the data already accessible in the data set. Note that compared to section 4.2 the models are less scattered in terms of the 10-fold CV errors with the synthetic data present in the model.

Tables 4 and 5 present the in sample and out of sample Poisson deviance for the best performing models in the case with simulated insurance data and with creating synthetic observations. There are not large differences between the models but there is definitely a benefit of applying more complex models when data is more balanced indicating that both customers and companies might benefit from better data usage and pre-processing of data. Interestingly GLM did not appear in the top models with data being more balanced.

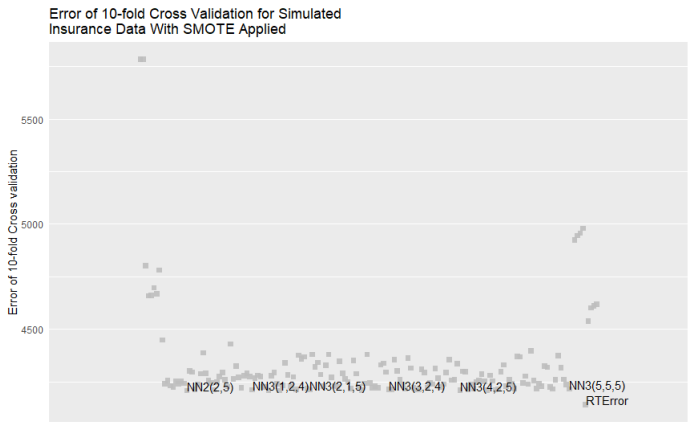

Figure 10: Cross validation errors of claims frequency for 168 different models applied on the Simulated data with SMOTE algorithm applied. "tot" indicates no grouping of continuous,"g(number)" indicated continuous variable has been grouped by interval of "number". "RF" is abbreviation for "Random Forest", "RT" is abbreviation for "Regression Tree" and "NN(n,k)" are Neural Networks with "k" hidden layers with "n" and "k" neurons in each layer.

	RT		NN(2,5)		NN(1,2,4)	
	In Sample	**Out of Sample**	**In Sample**	**Out of Sample**	**In Sample**	**Out of Sample**
1	25726,974	2578,214	45062,136	4779,904	48694,279	5417,677
2	25756,626	2588,730	26202,029	2923,042	37961,375	4195,913
3	25760,520	2603,277	33463,150	2813,038	45601,523	5056,003
4	25746,163	2646,892	26015,122	2008,504	55724,241	6152,621
5	25769,545	2640,172	19294,935	1524,747	23493,449	2630,592
6	25789,178	2594,702	36884,883	3128,732	27821,479	3069,298
7	25703,500	2621,138	23902,662	1860,521	49339,464	5503,973
8	24994,857	3381,858	28435,373	2419,146	41013,897	4558,548
9	21511,589	9246,025	35063,887	2983,485	28722,056	3225,044
10	20721,553	11297,514	48245,657	18141,801	26185,989	2925,688
Average	**24748,050**	**4219,852**	**32256,983**	**4258,292**	**38455,775**	**4273,536**

Table 4: Part 1 of the in sample and out of sample Poisson deviance of the 5 best performing models based on the 10-fold CV from out of sample predictions for the simulated data set with SMOTE algorithm applied. In sample and out of sample deviance's are presented for each of the folds from the 10-fold CV as well as the average of the CV.

	NN(2,1,5)		NN(4,2,5)	
	In Sample	**Out of Sample**	**In Sample**	**Out of Sample**
1	20128,086	2240,360	25598,234	2852,135
2	28771,925	3167,124	18901,915	2072,396
3	57064,535	6327,241	29875,532	3299,445
4	43330,944	4780,237	41511,785	4578,784
5	38165,836	4266,143	44432,657	4966,182
6	23179,560	2553,854	37638,445	4150,206
7	53157,436	5928,068	53286,284	5943,161
8	60573,690	6735,514	45826,016	5107,407
9	42586,340	4765,789	53940,111	6032,998
10	19470,642	2178,679	29956,480	3347,302
Average	**38642,900**	**4294,301**	**38096,746**	**4235,002**

Table 5: Part 2 of the in sample and out of sample Poisson deviance of the 5 best performing models based on the 10-fold CV from out of sample predictions for the simulated data set with SMOTE algorithm applied. In sample and out of sample deviance's are presented for each of the folds from the 10-fold CV as well as the average of the CV.

4.4 Wasa Motorcycle Insurance Data Without SMOTE

Figure 11 shows the plot of the cross validation error of claims frequency for each of the models trained on their test data. The labels of the 10 models with the lowest cross validation level on the are indicated on the plot. The RTG5 (regression tree with variables group as intervals of 5) model gives int the case the smallest deviation from the frequency meaning that the RTG5 manages to capture the underlying risk better than the other models. The other models are somewhat scattered around having various errors. Note her that the bulk of the data was simulated such that the most of the claims observations were observed where there risk was particularly well behaved giving the best performance for models which capture well behaved properties good. Note also that the scale of the errors is not excessively large. The performance of the error is measured on data which the models have not seen and more complex will have problems with

over-fitting and will have trouble performing on unseen extremes in the data.

Tables 6 and 7 present the in sample and out of sample Poisson deviance for the claims frequency in the data set for each of the folds in the 10-fold cross validation procedure. GLM has the lowest out of sample data deviance but not the lowest in sample deviance as the Regression Tree without grouping of the continuous variables has a lower in sample deviance. Interestingly the average in sample deviance of the neural networks seem to be higher compared to the other models but with a smaller difference in the out of sample performance.

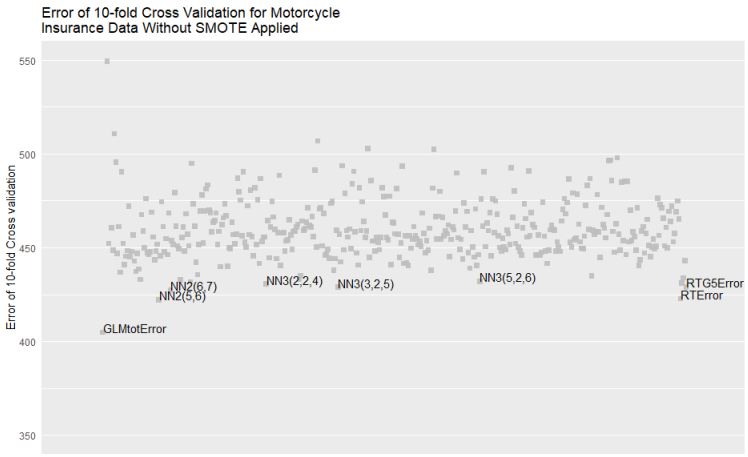

Figure 11: Cross validation errors of claims frequency for 410 different models applied on the Wasa Motorcycle Data without any SMOTE algorithm applied. "tot" indicates no grouping of continuous,"g(number)" indicated continuous variable has been grouped by interval of "number". "RF" is abbreviation for "Random Forest", "RT" is abbreviation for "Regression Tree" and "NN(n,k)" are Neural Networks with "k" hidden layers with "n" and "k" neurons in each layer.

	GLM		Regression Tree (w.o. grouping)		Random Forest (grouping interval of length 5)		NN(5,6)	
	In Sample	**Out of Sample**	**In Sample**	**Out of Sample**	**In Sample**	**Out of Sample**	**In Sample**	**Out of Sample**
1	3590,854	397,032	3576,353	419,528	4121,631	443,281	3996,428	422,560
2	3668,115	331,488	3679,610	316,552	4291,811	273,100	4592,085	430,703
3	3577,146	411,540	3590,620	409,590	4091,135	473,776	5416,510	384,101
4	3535,255	451,667	3497,633	469,948	4074,610	490,302	5164,117	251,768
5	3538,184	440,577	3535,683	462,772	4038,655	526,256	5146,010	501,268
6	3642,546	349,045	3683,340	366,717	4153,309	411,602	4890,586	527,409
7	3502,037	487,953	3551,583	525,154	3860,013	704,899	4551,993	508,875
8	3593,209	392,761	3509,431	391,542	4149,031	415,880	5462,625	505,420
9	3579,455	404,206	3618,402	406,601	4119,775	445,136	5074,039	478,804
10	3633,612	353,310	3586,512	367,746	4184,231	380,680	5332,250	260,811
Average	**3586,041**	**401,958**	**3582,917**	**413,615**	**4108,420**	**456,491**	**4962,664**	**427,172**

Table 6: Part 1 of the in sample and out of sample Poisson deviance of the 8 best performing models based on the 10-fold CV from out of sample predictions for the Wasa motorcycle insurance data set. In sample and out of sample deviance's are presented for each of the folds from the 10-fold CV as well as the average of the CV.

	NN(6,7)		NN(3,2,5)		NN(2,2,4)	
	In Sample	**Out of Sample**	**In Sample**	**Out of Sample**	**In Sample**	**Out of Sample**
1	4018,204	426,482	4674,467	463,185	4664,399	501,376
2	4892,104	436,486	5143,674	442,505	5188,266	442,814
3	4886,550	373,565	5877,059	393,070	5630,738	388,321
4	4612,126	230,909	6074,270	282,444	6173,129	285,955
5	4664,478	455,804	5546,145	493,867	5647,108	559,856
6	5426,586	575,145	5396,362	469,993	5344,586	505,974
7	4784,758	577,094	5201,118	508,062	5336,007	481,386
8	4395,758	534,893	5627,640	498,665	5525,130	453,085
9	4669,312	464,403	5782,013	480,545	5376,447	462,784
10	5004,815	244,676	6150,708	291,077	6135,545	290,869
Average	**4735,469**	**431,946**	**5547,346**	**432,341**	**5502,136**	**437,242**

Table 7: Part 2 of the in sample and out of sample Poisson deviance of the 8 best performing models based on the 10-fold CV from out of sample predictions for the Wasa motorcycle insurance data set without SMOTE applied. In sample and out of sample deviance's are presented for each of the folds from the 10-fold CV as well as the average of the CV.

4.5 Wasa Motorcycle Insurance Data With SMOTE

Applying first the SMOTE data to include a larger proportion of claims changes the picture of the best performing model drastically as can be seen in Figure 12 . The best model with the lowest 10-fold cross validation error being the random forest specifications. Also the neural networks performs in the top 10 of the 400 different specifications. Balancing out the data became more difficult for the GLM models to find a generalization of the underlying trend and the non-linear models performed better based on the 10-fold CV error. Especially the random forest with the grouped continuous variables with groups of 5. Comparing to the rest of the models performances there is a l difference of approximately 1500 in deviance of the out of sample error to the best performing model.

Table 8 summarizes the out of sample and in sample Poisson deviance for the

Wasa motorcycle data with SMOTE algorithm applied. On the contrary to the case when no SMOTE was not applied, the best performing is the Random Forest together with neural networks of various specifications. Both in sample deviance and out of sample deviance seem to agree on which of the best modes performs the best. It is important to note that the Random Forest is for the grouping of the variables which can be regarded as continuous in groups of 5.

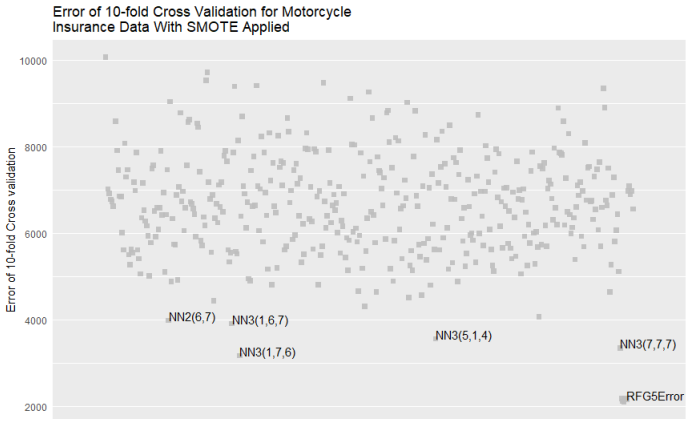

Figure 12: Cross validation errors for 410 different models applied on the Wasa Motorcycle Data with SMOTE algorithm applied. "tot" indicates no grouping of continuous,"g(number)" indicated continuous variable has been grouped by interval of "number". "RF" is abbreviation for "Random Forest", "RT" is abbreviation for "Regression Tree" and "NN(n,k)" are Neural Networks with "k" hidden layers with "n" and "k" neurons in each layer.

	RF (grouping interval 5)		**NN(6,7)**		**NN(1,7,6)**		**NN(7,7,7)**		**NN(5,1,4)**	
	In Sample	**Out of Sample**	**In Sample**	**Out of Sample**	**In Sample**	**Out of Sample**	**In Sample**	**Out of Sample**	**In Sample**	**Out of Sample**
1	20113,115	2543,333	43583,312	4756,914	33316,679	3634,824	31662,589	3453,836	27561,409	3096,340
2	20721,266	1935,182	16721,163	1845,358	9341,709	986,749	10992,408	1210,286	32339,571	3452,502
3	20538,917	2117,532	37811,732	4233,372	30217,968	3362,993	28135,716	3144,691	34152,411	3875,772
4	20112,608	2543,840	32195,199	3570,006	31380,612	3553,561	38817,222	4384,006	28989,096	3134,294
5	20333,593	2322,855	45504,745	5099,110	31107,976	3426,575	25381,115	2871,400	28192,537	3242,873
6	20635,213	2021,236	28584,536	3569,373	27593,555	3522,359	38057,707	4856,650	35512,177	4339,555
7	19755,002	2901,446	31736,420	3713,372	31893,397	3762,696	37211,935	4386,795	27774,054	3113,875
8	20481,973	2174,475	41476,526	4242,016	30323,677	3072,816	28039,420	2867,633	32233,558	3436,651
9	20489,833	2166,615	44179,090	4932,655	31179,014	3508,889	32747,831	3664,416	31151,712	3436,513
10	20726,515	1929,933	48313,359	5107,647	34198,606	3636,109	28806,258	3041,208	28135,587	2972,205
Average	**20390,803**	**2265,645**	**37010,608**	**4106,982**	**29055,319**	**3246,757**	**29985,220**	**3388,092**	**30604,211**	**3410,058**

Table 8: In sample and out of sample Poisson deviance of the 5 best performing models based on the 10-fold CV from out of sample predictions with SMOTE algorithm applied for the Wasa motorcycle insurance data set. In sample and out of sample deviance's are presented for each of the folds from the 10-fold CV as well as the average of the CV.

5 Conclusion

The objective of the thesis is to investigate the impact of imbalanced data with high risk customers being underrepresented in the data in terms of number of claims as well as exposure. As comparison, synthetic data was generating by using a SMOTE algorithm.

The thesis show that given the imbalance in the data, more sophisticated models fail to outperform standard models such as the GLM meaning that the price will be more or less smoothed given that the claim cost is stable which can be seen in Figure 9 and Figure 11 as well ass tables 2,3,6 and 7. However, when synthetic observations applying the SMOTE method were included, GLM failed to perform better than the models which are designed to fit the data better as summarized by the 10-fold cross validation in Figure 10 and Figure 9 and tables 4,5 and 8.

This is not surprising as even if there exist high risk customers in the portfolio but with a relatively small proportion of the portfolio, their outcome will have a small weight and effect of the prediction error costs relatively less despite perhaps having larger claims than actual. Including more data however, shifted the errors and models which were able to capture more complex structures in the data performed better than the GLM (Figure 10, Figure 12). This shows that data balancing, a situation which is difficult to attain without additional expensive data collection or bold assumptions, is required to show that the modern machine learning type algorithms are performing better and to utilize their benefits in insurance pricing. There is however potential to formulate some form of tuning process more efficient than utilized in this thesis to discard the large number of models specified in order to find some optimal specification or number of trials.

Despite data being deliberately created such that there is a presence of non-linearity in section 4.2 it seemed as if the machine learning methods where quite sensitive to the non-linear structures having quite a weak signal. In section 4.3 on the other hand SMOTE was applied to over sample by 20% meaning that only an extra 20% of synthetic claims were used in the data. In a real world insurance pricing application there can be substantial consequences to the insurance providers profitability or portfolio health. A result of GLM performing the most will impact policyholders such that high risk customers will pay less than they need to and low risk more. If this can be mitigated applying SMOTE and machine learning predictive techniques there is a big potential for the method.

There are a few areas where the thesis could be extended and improved. First of all the complexity of the mu parameters of simulated data could have been increased gradually in combination with shifting the exposure with data im-

balance to investigate potential threshold where the performance of the model shift fast. This could give insight to pricing professionals of the level of balance needed in the data to use more complex methods in their insurance pricing. Further, the specification of the SMOTE algorithm could be investigated closer as well as other data balancing methods in order to find both the level at which there are large shifts in model performance. Comparisons with simulated balanced data are interesting and could be evaluated but since it is not a realistic situation it is unlikely of interest. The comparison could be done to evaluate the fit of the training data rather than comparing with the test data using the 10-fold CV technique. In realistic situations, a insurance stock will be somewhat static with the new unseen data being marginal compared to the rest of the portfolio where it will be more of interest in updating the prices of the current stock. Instead of the approach of balancing the data, a Bayesian approach on the simulated data could be used in the comparison to investigate if the model can be improved further as well as comparing the performance of the method to a Bayesian technique and over-sampling using SMOTE. Also there are the possibilities to work with different types of variable transformation and data split to improve the prediction properties of the method.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2016.

[2] Accenture. Machine Learning in Insurance. http://www.accenture.com, 2018. Accessed: 2021-02-16.

[3] Leo Breiman. *Classification and regression trees*. Chapman Hall, Boca Raton, Fla, 1984.

[4] Statistiska Centralbyrån. Folkmängden efter region, civilstånd, ålder och kön. År 1968 - 2020, 2021. Accessed 2021-02-15.

[5] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[6] Iain D. Currie. On fitting generalized linear and non-linear models of mortality. *Scandinavian actuarial journal*, 2016(4):356–383, 2016.

[7] H. B. Curry and I. J. Schoenberg. On pólya frequency functions iv: The fundamental spline functions and their limits. *Journal d'analyse mathématique (Jerusalem)*, 17(1):71–107, 1966.

[8] Charles Dugas, Yoshua Bengio, Nicolas Chapados, Pascal Vincent, Germain Denoncourt, and Christian Fournier. Statistical learning algorithms applied to automobile insurance ratemaking. In *CAS Forum*, volume 1, pages 179–214. Citeseer, 2003.

[9] Andrea Gabrielli, Ronald Richman, and Mario V. Wüthrich. Neural network embedding of the over-dispersed poisson reserving model. *Scandinavian actuarial journal*, 2020(1):1–29, 2020.

[10] J. Garrido, C. Genest, and J. Schulz. Generalized linear models for dependent frequency and severity of insurance claims. *Insurance, mathematics economics*, 70:205–215, 2016.

[11] Markus Gesmann, Dan Murphy, Y Zhang, Alessandro Carrato, G Crupi, M Wuthrich, and F Concina. Chainladder: Statistical methods and models for claims reserving in general insurance. *URL: CRAN. R-project. org/package= ChainLadder*, 2015.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. The MIT Press, Cambridge, Massachusetts, 2016.

[13] Daniel Graupe. *Principles of artificial neural networks*, volume 6.;6;. World Scientific, Hackensack, N.J;Singapore;, 2nd edition, 2007.

[14] Robert M. Gray. *Probability and Random Processes*. Springer New York, New York, NY, 1988.

[15] Trevor J. Hastie and Robert J. Tibshirani. *Generalized additive models*, volume 43. Chapman and Hall, London, 1. edition, 1990.

[16] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning: with applications in R*, volume 103. Springer, New York, NY, 2017;.

[17] Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas, et al. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36, 2006.

[18] Kevin Kuo. Deeptriangle: A deep learning approach to loss reserving. *Risks*, 7(3), 2019.

[19] P. McCullagh. *Generalized linear models*. Routledge, 2019.

[20] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT Press, Cambridge, MA, 2012.

[21] Karl P Murphy, Michael J Brockman, and Peter KW Lee. Using generalized linear models to build dynamic pricing systems. In *Casualty Actuarial Society Forum, Winter*, pages 107–139. Citeseer, 2000.

[22] Esbjørn Ohlsson and Bjørn Johansson. *Non-life insurance pricing with generalized linear models*. Springer, Heidelberg;New York;, 2015;.

[23] Pietro Parodi. *Pricing in general insurance*. CRC Press, Boca Raton, Florida;London, [England];New York, [New York];, 1 edition, 2015;.

[24] ASTIN Big Data /Data Analytics Working Party. Phase 1 Paper - April 2015. http://www.actuaries.org, 2015. Accessed: 2021-02-16.

[25] T. Pentikäinen R. E. Beard and E. Pesonen. *Risk Theory*. Chapman Hall, 1984.

[26] Arthur E Renshaw. Modelling the claims process in the presence of covariates. *ASTIN Bulletin: The Journal of the IAA*, 24(2):265–285, 1994.

[27] Matthew S Spencer, Samantha C Bates Prins, Margaret S Beckom, et al. Heterogeneous distance measures and nearest-neighbor classification in an ecological setting. *Missouri Journal of Mathematical Sciences*, 22(2):108–123, 2010.

[28] Luís Torgo, Rita P. Ribeiro, Bernhard Pfahringer, and Paula Branco. *SMOTE for Regression*, pages 378–389. Springer Berlin Heidelberg, Berlin, Heidelberg.

[29] Mario V Wuthrich and Christoph Buser. Data analytics for non-life insurance pricing. *Swiss Finance Institute Research Paper*, (16-68), 2019.

[30] Mario V. Wüthrich. Machine learning in individual claims reserving. *Scandinavian actuarial journal*, 2018(6):465–480, 2018.