Towards Machine Learning: Alternative Methods for Insurance Pricing – Poisson-Gamma GLM's, Tweedie GLM's and Artificial Neural Networks

UC Berkeley
Statistics Honors Thesis

By

Navarun Jain

Advisor: Dr. David Brillinger

Abstract

This thesis discusses three approaches to pricing insurance – Poisson-Gamma GLM's for claim frequency and claim severity, Tweedie Compound Poisson GLM's and Artificial Neural Networks. A brief mathematical and theoretical background of each model is discussed, along with an explanation of the underlying processes and relevant hyperparameters. 4 approaches are presented to asses and compare each model – test data MSE, residual plot analysis, AIC, 5-fold cross-validation and risk premium ratio analysis to determine which groups of policyholders carried the most risk. The models were trained and tuned on a one-year vehicle damage insurance claims dataset, and optimal values were found for Tweedie and Neural Network hyperparameters. It was found that the Poisson-Gamma GLM was the most accurate, but only in terms of test data MSE. In all other approaches, the Tweedie GLM and the Neural Network were found to be comparable and, in some cases, better than the Poisson-Gamma GLM. In terms of goodness-of-fit, all models were comparable. Overall, it was concluded that autonomous machine learning algorithms such as Neural Networks hold great potential for actuaries in insurance ratemaking.

Acknowledgements

I would like to sincerely thank my advisor, Dr. David Brillinger, for his guidance and support in helping my write this thesis. It is truly a great honor for me to have been mentored and advised by as brilliant as him. I would also like to thank Shivash Bhagaloo, Raghav Ohri and the team at Lux Actuaries & Consultants for having given me a platform and the encouragement to begin my journey in the field of applied machine learning in the actuarial field.

I would also like to thank my Mum, Dad and *Nani* for having supported me in all my endeavors and for always believing in my and having my back.

1. Introduction

Today's age is that of big data. Data rules every decision that is taken in almost every field, be it economics, education or engineering. Insurance is also no stranger to this fact. It has now become much easier to collect, record and transport large datasets. Further, with increasing computing power and the development of statistical modeling tools and techniques, it has now become easier to analyze such large datasets and draw meaningful conclusions from them. This is something the insurance industry can really rely on. Due to the increased volume of data, it is now possible to asses risks more comprehensively and, thus, make better decisions and predictions on how to mitigate the same.

A key issue in insurance ratemaking is that if premiums are too high, consumers will turn to other companies to purchase insurance. However, if premiums are too low, companies will not earn enough premiums to cover claims. Further, risk characteristics must be chosen appropriately such that it is reasonable to charge different groups of consumers different rates based on them. In deciding rates, companies need to have an adequate estimation of the expected amount a policyholder might claim in case of an accident. Hence, for example, it would make sense to charge someone owning a new Rolls Royce a much higher premium for auto insurance than someone owning a used Toyota Corolla. Thus, one of the biggest challenges that has faced the insurance sector, and particularly actuaries, is – how do we comprehensively, fairly and adequately price insurance products taking into account a given set of risk characteristics of

policyholders? How do we make sure that the rates adequately mirror the amount of risk that these characteristics may present?

The introduction of Generalized Linear Models, or GLM's, has provided an answer to this. However, as is outlined later in this thesis, GLM's require us to have some knowledge of the underlying patterns in the data. Actuaries are realizing the need to automate the process by which underlying patterns and/or anomalies are detected in data and meaningful predictions are generated on the basis of these detected patterns. This is leading to a resurgence of interest in Artificial Neural Networks as a ratemaking tool. With increased computing power and better deep learning libraries, it is becoming easier to train neural networks that are efficient and robust. This thesis explores the underlying theory behind GLM's and Artificial Neural Networks and discusses the applications of these in pricing auto insurance.

2. Theoretical and Mathematical Background

2.1 Expected Risk Premium and Gross Premium

Let's assume a policyholder has *n* risk characteristics, such as age, gender, vehicle type and vehicle age. Then, his/her specific set of risk factors would be given by a $1 \times n$ vector. Let $X \in R^n$ be the space of all such vectors. The process of insurance ratemaking aims to determine a policyholder's expected claim cost, which is known as the *expected risk premium*. Let P be the expected risk premium for a given policyholder. Then, given a set of risk characteristics, we have

$$f(x) = E[P|X = x]$$

where x corresponds to the $1 \times n$ vector for this policyholder, the observations of which contain the policyholder's specific set of risk characteristics. The goal of statistical learning is to approximate the function $f(x)$ that can model this expected risk premium.

This risk premium, however, is only one part of the overall rate that policyholders pay. The overall rate, known as *Gross Premium,* comprises of the risk premium along with loadings for profit, administrative, business and expected loss adjustment expenses.

2.2 Generalized Linear Models (GLM's)

GLM's are an extension of the classical linear model, which is defined by the following equation:

$$y_i = \sum \beta_k x_{ik} + \epsilon_i \ (\forall i \in [1, n])$$

where *n* is the number of observations in the data.

A GLM consists of three components (Fox 2016) –

- A *random component*, which specifies the conditional distribution of the response variable y_i (i^{th} of *n* observations), given the values of the explanatory variables. Thus, the random component specifies the distribution of $E[y_i \mid \{x_1, \dots, x_k\}]$, where *k* is the number of explanatory variables in the data. This distribution usually belongs to the exponential family, which includes distributions such as Gaussian (normal), binomial, Poisson, gamma or the inverse Gaussian families.
- A *linear predictor*, also known as the *systematic component*, which is given by

$$\eta_i = \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_k X_{ik}$$

- A smooth and invertible *linearizing link function*, which relates the random component to the systematic component. In other words, the link function transforms the expected value of the response variable, $\mu_i = E[y_i \mid \{x_1, \ldots, x_k\}$, to the linear component. Thus, we have

$$g(\mu_i) = \eta_i = \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_k X_{ik}$$

Since the link function is invertible, we get

$$\mu_i = g^{-1}(\eta_i)$$

As a result, a GLM can be interpreted as a linear model of the transformation of the expected value of a response variable, or as a nonlinear regression model for the response variable.

An important and useful property of distributions in the exponential family is that the conditional variance of y_i is a function of its mean μ_i and, in some cases, a constant *dispersion parameter* ϕ (Fox 2016). The primary function of this dispersion parameter is to indicate the specific distribution that is used. Some common distributions, their associated link functions and their variance functions are summarized below.

Distribution	**Link Function**	**Variance Function**
Gaussian	Identity	ϕ
Binomial	Logit	$\frac{\mu_i(1 - \mu_i)}{n_i}$
Poisson	Log	μ_i
Gamma	Inverse	$\phi\mu_i^2$

Inverse Gaussian	Inverse square	$\phi\mu_i^3$

Table 1: Summary of Exponential Family Distributions. Here, $\mu_i = E[y_i \mid \eta_i]$ and n_i is the number of trials for the binomial distribution (Fox 2016).

This thesis discusses the two primary approaches to using GLM's in modeling risk premiums. These are the standard Poisson-Gamma model for claim frequency and severity, and the Tweedie compound Poisson model for expected losses.

2.2.1 Poisson-Gamma Model for Claim Frequency and Claim Severity

The standard approach consists of fitting 2 separate GLM's. The first is a model with Claim Frequency as the dependent variable having a Poisson distribution with a log link. The second is a model with Claim Severity as the dependent variable having a Gamma distribution with a log link (Andersen et al. 2004) Since the gamma distribution cannot take non-positive values, the severity model is only fitted on the subset of the data for which severity > 0.

Claim Frequency is given by the number of claims in a given period, usually one year. Hence, when Claim Frequency is modeled, the dependent variable is usually claim count. On the other hand, Claim Severity is the average loss associated with a single claim, i.e.

$$Severity = \frac{Total\ Claim\ Amount}{Claim\ Count}$$

In this approach, the Poisson and Gamma models are fitted on the data, and the final risk premium is predicted as follows:

$$Expected\ Risk\ Premium$$
$$= Expected\ Claim\ Frequency\ (from\ Poisson\ model)$$
$$\times Expected\ Claim\ Severity\ (from\ Gamma\ model)$$

2.2.2 Tweedie Compound Poisson model

A tweedie distribution is one which satisfies

$$V(\mu) = \mu^d$$

where *d* is a parameter that represents the distribution of the response variable, and μ is the mean of the response variable.

This approach assumes that expected claim costs follow a compound Poisson distribution (Smyth & Jørgensen 2002). The approach is as follows:

Assume N_i is the observed claim count for the i^{th} category and let Z_i be the observed claim amount for that category. Let w_i be the number of units at risk for the given category (let this be 1 policy year). Then, observed claim cost for this category, Y_i, is given by

$$Y_i = \frac{Z_i}{w_i} = Z_i$$

since we assume $w_i = 1$. Thus, we are now directly modeling claim cost instead of modeling frequency and severity separately.

We assume that that N_i follows a Poisson distribution with mean λ_i, and claim size follows a Gamma distribution with mean τ_i and shape parameter α (Smyth & Jørgensen 2002). Them the conditional distribution of Y_i given N_i also follows a Gamma distribution with mean $N_i\tau_i$, whenever $N_i > 0$.

The Tweedie approach assumes that $\mu_i = E(Y_i) = \lambda_i\tau_i$, with the variance of the response variable follows an exponential distribution as μ_i varies. Hence, we have

$$V(Y_i) = \phi\mu_i^p$$

where *p* is the distribution parameter (also referred to as the variance power parameter), given in terms of the Gamma shape parameter as

$$p = \frac{\alpha + 2}{\alpha + 1}$$

This implies that, since $\alpha > 0$, we must have $1 < p < 2$ for the distribution to be compound Poisson.

Using the conditional variance method, the variance of Y_i can be calculated directly as

$$V(Y_i) = E_{N_i}V(Y_i|N_i) + V_{N_i}E(Y_i|N_i) = \left(\frac{1}{\alpha} + 1\right)\lambda_i\tau_i^2$$

Equating this to the existing formula for the variance, we get

$$\phi\mu_i^p = \left(\frac{1}{\alpha} + 1\right)\lambda_i\tau_i^2$$

Now, we know,

$$p = \frac{\alpha + 2}{\alpha + 1}$$

$$\Longrightarrow \alpha = \frac{2 - p}{p - 1}$$

This gives

$$\phi\mu_i^p = \frac{1}{2-p}\mu_i\tau_i^2$$

$$\Longrightarrow \phi = \left(\frac{1}{2-p}\right)\mu_i^{1-p}\tau_i^2$$

Using the fact that $\mu_i = \lambda_i\tau_i$, we get

$$\phi = \left(\frac{1}{2-p}\right)\lambda_i^{1-p}\tau_i^{2-p}$$

Thus, we can estimate the dispersion parameter as a function of λ_i, τ_i and the variance power parameter p (Smyth & Jørgensen 2002).

2.2.3 Offsets

In GLM theory, an offset is a variable whose coefficient is constrained to 1 (Yan et al. 2009). The standard model is given by

$$Y = \beta X + \epsilon$$

where X is an $n \times (k+1)$ matrix of the input variables, and β is the matrix of model coefficients. When an offset is added to the model, the above equation is modified as follows –

$$Y = \beta X + \zeta + \epsilon$$

Here, ζ is the offset variable added (Yan et al. 2009).

In a GLM, the response variable is transformed by the link function to linearize an otherwise multiplicative model. Hence, the offset needs to be in the same scale as the model variables. For example, if a logarithmic link is used, we get

$$\log(E[Y]) = \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_k X_{ik} + \log(\zeta)$$

where ζ represents the offset variable. Since we use a logarithmic link function, the offset variable is also log-transformed.

2.3 Artificial Neural Networks

Artificial Neural Networks (ANN's) are a type of computing system built based on the construction of the human brain. Just as the human brain learns and remembers by pattern matching and can apply these learned patterns to new situations by association, ANN's too can detect patterns within data and apply these patterns to predict new cases. An ANN is a structured sequential model. It consists of an input layer, one or multiple hidden layers, and an output layer. Each layer comprises of neurons, which are the basic computing units in an ANN model. Neurons from each layer are connected to those from the next layer, thereby creating a network structure. Each neuron past the input layer receives data from the previous layer as a weighted sum of outputs from the previous layer. Each neuron in the input layer represents an input variable, and each neuron in the output layer represents an output variable. Thus, in the case of regression, there is one neuron in the output layer.

Data that comes into every neuron is transformed by an activation function. The result is then passed on to the next layer. Thus, in an ANN, every neuron in the hidden layer is a linear combination of the outputs from the previous neuron, transformed by the activation function as follows (Kuhn & Johnson 2016) –

$$h_k(x) = g(\beta_{0k} + \sum_{i=1}^{n} x_i \beta_{ik})$$

where $h_k(x)$ represents the k^{th} neuron in a hidden layer, and β_{ik} represents the coefficient of the i^{th} previous-layer neuron on the k^{th} neuron in a hidden layer. In the case of the first hidden layer, this represents the effect of the i^{th} predictor on the k^{th} neuron in a hidden layer. Here, $g(\cdot)$ represents the activation function, which transforms the linear combination of inputs from one layer and outputs it to the next layer. A sample ANN structure is shown below.

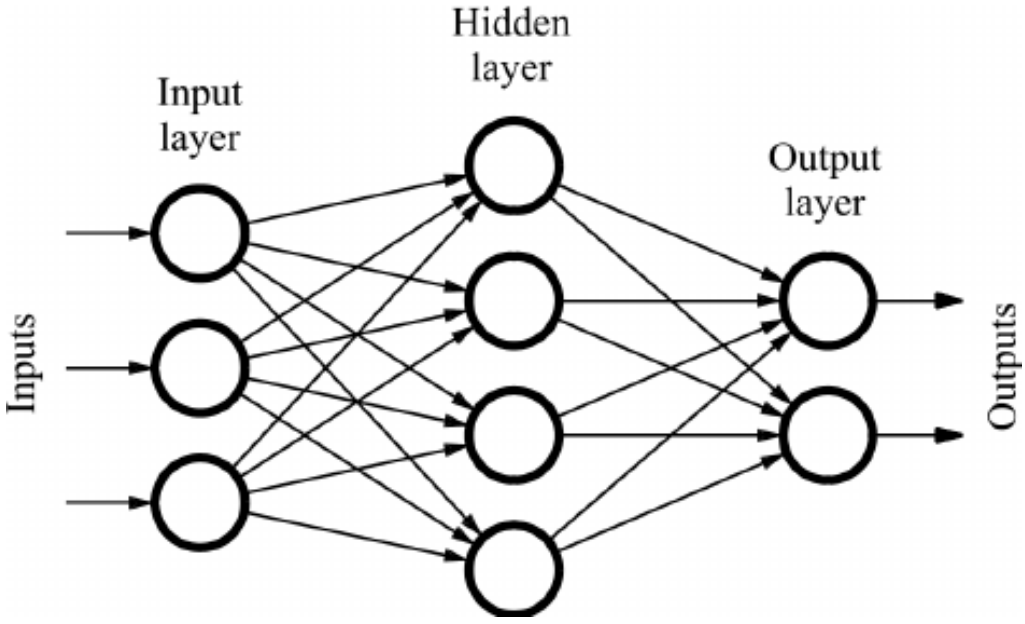

Figure 1: A sample ANN structure. In this model, information flows from the input layer to the output layer in the form of connections between neurons. Since information only flows in one direction, this type of ANN is called a Feed Forward ANN.

The activation function should be bounded, monotonic, continuous and differentiable everywhere. Some examples of good activation functions are summarized below.

Function	$\boldsymbol{g(x)}$
Sigmoid	$\frac{1}{1+e^{-x}}$
Hyperbolic Tangent	$\frac{e^{-2x}-1}{e^{-2x}+1}$
Gaussian	$e^{-\frac{x^2}{2}}$
Identity	x

Table 2: Some commonly used ANN activation functions

2.3.1 Backpropagation and Loss Optimization

The weights in an Neural Network model are typically learned by the backpropagation algorithm, which is a 2-stage iterative process. At first, the weights are randomly initialized. In the forward propagation stage, the training data is passed into the model, and the randomly initialized weights are used to generate predictions from the model. These predictions are then compared with the observed values of the response variable. This comparison is done based on a pre-defined L. This could be the any metric that compares the deviation between the predicted and observed values of the response variable in the training data, such as the training MSE.

In the second stage, the weights are revised such that the training error is lowered. Thus, the model tries to minimize the gradient of the loss, given by

$$\nabla L = \frac{\delta L}{\delta W}$$

Here, L is the loss function (such as the training MSE) and W is the weights and bias parameters in the network (Rumelhart et al. 1986). The model stops when one of two things are attained – either a minimum is reached in the gradient function above, or the model runs for a pre-determined number of iterations, or epochs, at which point it is deliberately stopped.

2.3.2 Optimization by Stochastic Gradient Descent

Stochastic Gradient Descent is an optimization technique implemented to make the learning algorithm run faster. In this process, the training error is not calculated on the entire training dataset at each iteration. Instead, the loss function is computed for one randomly selected observation, and a step is taken in the negative direction of the gradient with respect to the selected observation (Hastie et al. 2009). Thus, the weights are revised in the direction in which the loss function is minimized the most. The size of the step taken is known as the *learning rate*. This is one of the most important hyperparameters in an ANN. The learning rate gives us an estimate of how drastically the model should "change its mind" in revising the weights in the direction in which the loss function is minimized the most. The higher its value, the larger the step taken in this direction.

Another way to make the model train faster is to run it on small subsets, or *batches*, of the training data at every iteration instead of training it on the entire training data. The batch size to be taken is another hyperparameter that can be tuned.

3. Model Assessment and Comparison

3.1 Test MSE

The model is run on the entire training dataset, and predictions are generated for the test dataset. The observed values and predicted values are compared for the test dataset by computing the Mean Squared Error (MSE) on the test dataset, as follows.

Let Y_i be the observed value of the response variable and let $\widehat{Y}_i$ be the predicted value. Then, the test MSE is given by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{Y}_i - Y_i\right)^2$$

The test MSE's are computed for every model trained on the data. Since the MSE gives an indication of the predictive power of the model, the model with the lowest test MSE is the most accurate.

3.2 Akaike Information Criterion (AIC)

AIC is given by the formula

$$AIC = -2 \times \log(\mathcal{L}(\hat{\theta}|y) + 2K$$

where $\mathcal{L}(\hat{\theta}|y)$ is the likelihood function for the distribution of the model parameters, given the data, and *K* is the number of model parameters, which is given by one plus the number of explanatory variables passed into the model. The AIC gives an estimate of the expected relative distance between a fitted model and the unknown true mechanism that actually generated the observed data (Burnham & Anderson 2002). In other words, the AIC gives an estimate of the goodness of fit of a model. The lower the AIC, the better the model can explain the variation in the data.

The AIC can also be approximated by using the residual sum of squares as

$$AIC = n \times \ln\left(\frac{RSS}{n}\right) + 2K$$

where RSS is the sum of the squared residuals from the model, and *n* is the number of rows in the test data (Panchal et al. 2010).

3.3 Risk Premium Ratios

The risk premium ratio is given by

$$\frac{Observed\ Claims\ Cost}{Expected\ Risk\ Premium}$$

where the Observed Claims Cost comes from the dataset, and the Expected Risk Premium is predicted by the model. For every model, this is calculated over the entire dataset, in order to determine whether or not the rates are actuarially fair, from a profitability and adequacy point of view. The risk premium ratio is also calculated for every risk factor in the dataset, where the values are compared between categories in order to determine which risk groups are riskier

than others. It is expected that all models will more or less show the same patterns in terms of mirroring the differences in risk between different categories of rating factors.

3.4 Cross-Validation

In this iterative approach, the data is randomly divided into *k* groups, called folds. The folds are created such that no 2 folds contain the same data points. In other words, the folds are created such that there is no correlation between them. In every iteration, one of the folds is left out as the validation set, and the model is run on the remaining *k – 1* folds. (James et al. 2013) The MSE is then computed for the fold that is left out, as is described above. This process is run *k* times, where, in every run, one of the folds is left out. The k-fold cross validation error is then computed as the mean of the errors computed for each fold, i.e.

$$MSE_{cv} = \frac{1}{k}\sum_{j=1}^{k} MSE_j$$

where

$$MSE_j = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{Y}_i - Y_i\right)^2$$

for the j^{th} fold.

Cross-validation gives us a good idea of how well a model would perform on an independent dataset (James et al. 2013). This is particularly useful when we are tuning the hyperparameters of a model, such as the variance power parameter in a Tweedie GLM, or the batch size and learning rate in an ANN.

4. Applications to Insurance Data

4.1 The Data

The data used contains policyholder-level information on one-year vehicle insurance policies, taken out in 2004 or 2005. It comes from the *insuranceData* package in R. It contains 67,856 rows, each of which corresponds to a unique policyholder. A description of the rating factors is outlined below.

Rating Factor	Description
veh_value	Vehicle value in $10,000's; a continuous variable
veh_body	Vehicle body type; a categorical variable that contains the following levels: BUS, CONVT, COUPE, HBACK, HDTOP, MCARA, MIBUS, PANVN, RDSTR, SEDAN, STNWG, TRUCK and UTE
veh_age	A categorical variable with levels 1, 2, 3 and 4. Level 1 represents the newest vehicles and 4 representing the oldest vehicles
gender	A categorical variable with 2 groups – F (Female) and M (Male); represents gender of driver/vehicle owner

area	A categorical variable representing the location of the vehicle owner; grouped into 6 categories – A, B, C, D, E and F
agecat	Age of vehicle owner grouped into 6 categories – 1, 2, 3, 4, 5 and 6, with 1 representing the youngest owners and 6 representing the oldest

Table 3: A description of the rating factors in the dataset

There are three response variables that are modeled in this dataset. The primary response variable is *claimcst0*, which gives the observed claim cost when there is a claim. The variable *numclaims* gives the claim count when there is a claim. If there is no claim, then both *claimcst0* and *numclaims* are 0. The third variable is *severity*, which is created in the dataset as per the formula

$$severity = \frac{claimcst0}{numclaims}$$

If there is no claim, then the value of *severity* is set to 0.

The distribution of claim costs is highly asymmetric, with almost all the values at 0, and very few values above 0, as is seen in the histogram below.

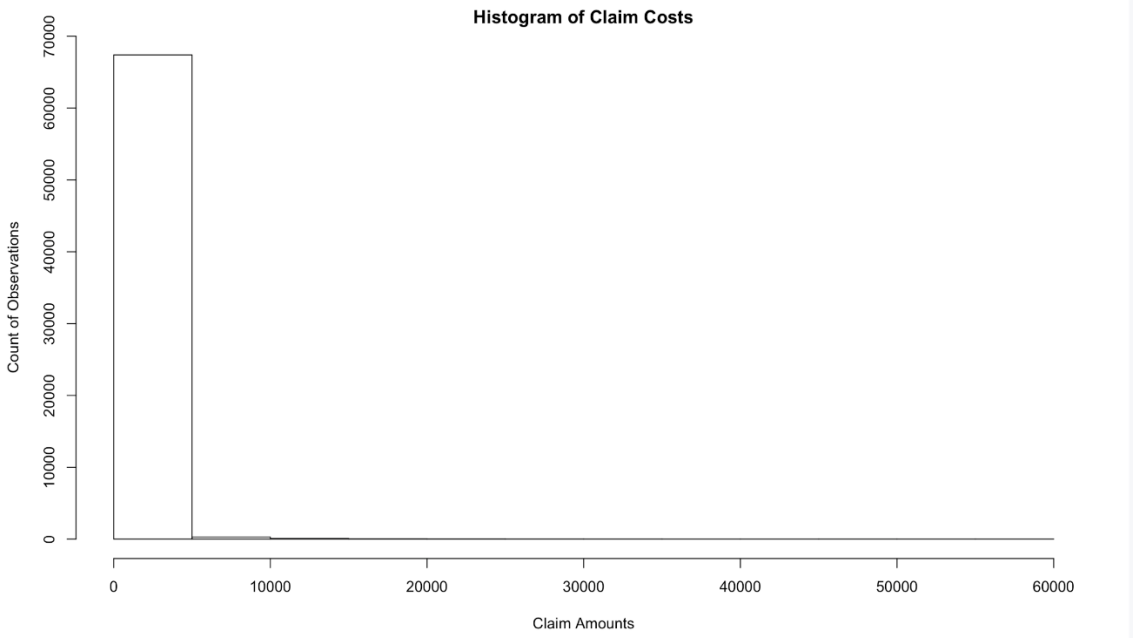

Figure 2: Histogram of Raw Claim Sizes

On the other hand, the distribution of the log of claim sizes seems to much more symmetric, although still slightly left skewed.

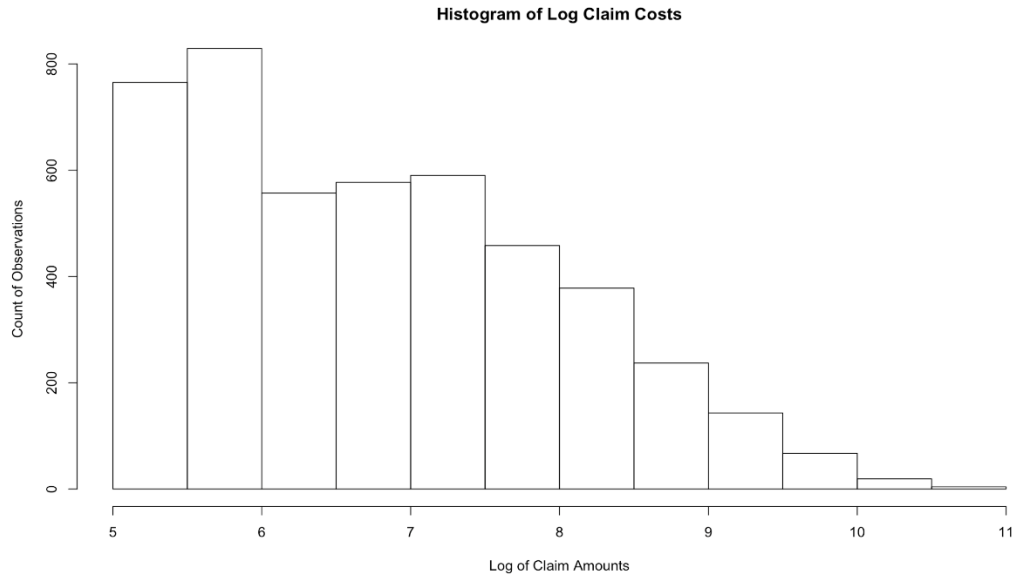

Figure 3: Histogram of log-transformed Claim Sizes

The data was aggregated over the rating factors outlined in Table 3, and the aggregated dataset contains 45,220 observations. 80% of the data was used as the training/validation dataset, and 20% was used as the test dataset.

4.2 The Models

4.2.1 Poisson-Gamma GLM

The Poisson model was fitted with *numclaims* as the response variable and the rating factors in Table 3 as the explanatory variables. A logarithmic link was used. The offset was set to the log of the variable *exposure*. Since *exposure* measures propensity to risk, it has a direct interaction with claim frequency, which cannot be modeled. It is a constant term that is added to the linear predictor without the term being estimated. Hence, its effect is added outside of the model as an offset.

The Gamma model was fit with *severity* as the explanatory variable and the rating factors outlined in Table 3 as the explanatory variables. The variable *exposure* was not used since it has no effect on the average loss per claim. A logarithmic link was used again.

4.2.2 Tweedie GLM

A log-link Tweedie compound Poisson GLM was fitted, with *claimcst0* as the explanatory variable. The log of *exposure* was added as an offset to take into account the claim frequency side of the compound model. For the variance power parameter, the following values were tested:

$$p = \{1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9\}$$

The optimal tweedie variance power parameter was found by maximum likelihood estimation. Using the formula described in Section 2.2.2, the algorithm computes the value of ϕ for each of the given values of p. It then computes the value of the log-likelihood function for each value of p, using the distribution approximated by ϕ, and uses these to estimate the optimal value of p. The results are shown below.

$\boldsymbol{p}$	$\boldsymbol{Estimted\ \phi}$	$\boldsymbol{Log-likelihood\ at\ p}$
1.1	816.24	-5.2×10^5
1.2	702.71	-4.57×10^5
1.3	524.23	-4.33×10^5
1.4	376.5	-4.22×10^5
1.5	271.47	-4.1755×10^5
1.6	201.81	-4.1764×10^5
1.7	180.52	-4.27×10^5
1.8	No convergence	No convergence
1.9	No Convergence	No Convergence

Table 4: Log-likelihood estimates of Tweedie variance power parameter and estimates of dispersion parameter

A smoothed plot of the log-likelihood values for different value of p is also shown below.

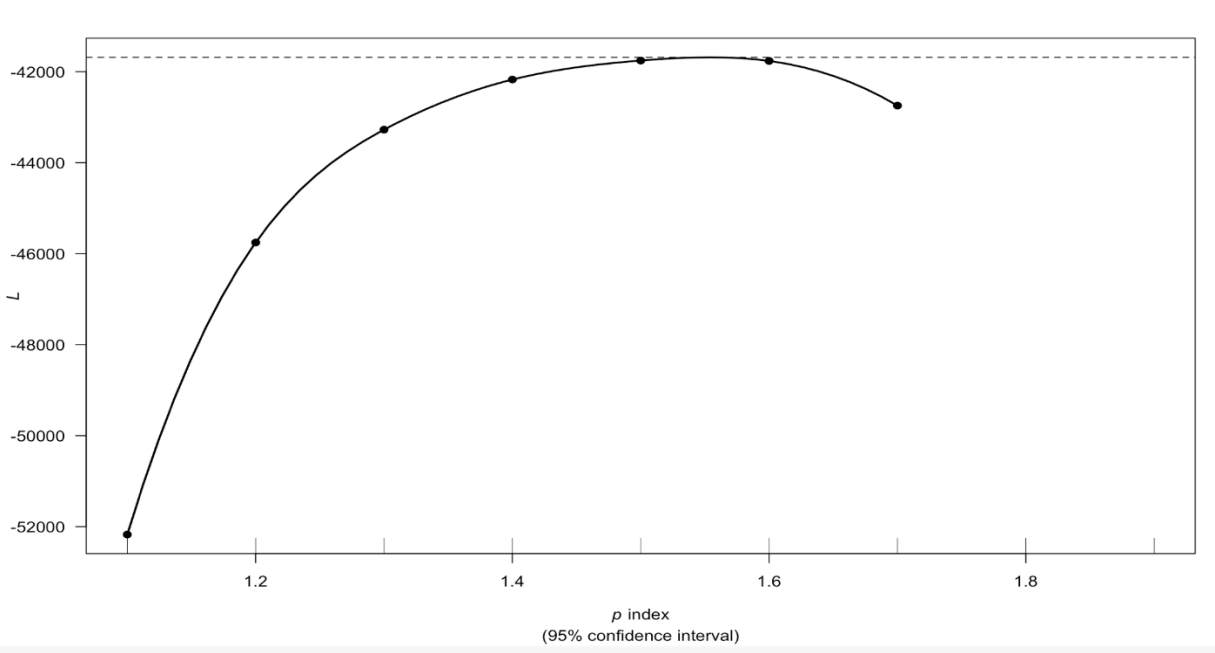

Figure 4: Plot of log-likelihood values for different value of p

The most optimal value of p was found to be 1.553. This was, thus, the variance power parameter used to train the Tweedie compound Poisson model with a log link.

4.2.3 Artificial Neural Networks (ANN's)

4.2.3.1 Data Preparation

There were two additional steps that were taken to prepare the data before training an ANN on it. Unlike GLM's, ANN models cannot handle categorical data. Hence, categorical variables had to be manipulated so that they are passed into the model as numbers. This was done by a process known as one-hot encoding, which transforms categorical variable into numerical variables to be used in machine learning algorithms to improve prediction accuracy and efficiency. Categorical variables were identified in the data, and dummy indicator variables were created for each unique

category in these variables. Thus, for example, if a certain row in the data has a vehicle body of BUS, the dummy indicator for *veh_body_BUS* was assigned a value of 1. Wherever the body type was not bus, this variable was assigned a value of 0. As a result of dummification, we now had 32 nodes in the input layer.

For this model, exposure was added as a predictor variable. Since we do not assume anything about the underlying distribution of the response variable and the patterns in the data, we use exposure as an input node rather than estimating it outside the model as an offset, as was done in the case of the GLM's. Thus, in total, we had 33 nodes in the input layer.

The next step in transforming the data was to normalize it. Since all the variables have difference scales, it is advisable to bring them all to one common scale in order to improve efficiency and prediction accuracy. The data was, thus, min-max scaled. As a result, all the data was in the range of (0, 1). This was done for each row according to the formula

$$x_i^{scaled} = \frac{x_i - x^m}{x^M - x^m}$$

where x^M and x^m are vectors representing the maximum and minimum values of each variable in the data, respectively, and x_i is a vector representing the i^{th} row in the data, i.e. the vector $[x_{i1}, x_{i2}, \dots, x_{ik}]$ where k is the number of predictor variables in the data. The predictions were scaled back by applying the inverse of the above formula to the same.

4.2.3.2 Choice of Activation Function

The sigmoid activation function was chosen to train the neural networks. The function is given by

$$f(x) = \frac{1}{1 + e^{-x}}$$

There are two reasons for choosing the sigmoid activation function. First, we do not expect any negative risk premiums. The range of the sigmoid function is (0, 1), which is best suited for positivity. This is also the motivation behind normalizing the data to the (0, 1) range.

The choice of the sigmoid activation function is also the result of Cybenko's Universal Approximation Theorem, which states that any real, continuous and bounded multivariate function can be approximated by a feed forward ANN with one hidden layer and a sigmoidal activation function, provided that there are no constraints placed on the number of nodes and the size of the weights (Cybenko 1989). The general form of a sigmoidal activation function is given by the following graph.

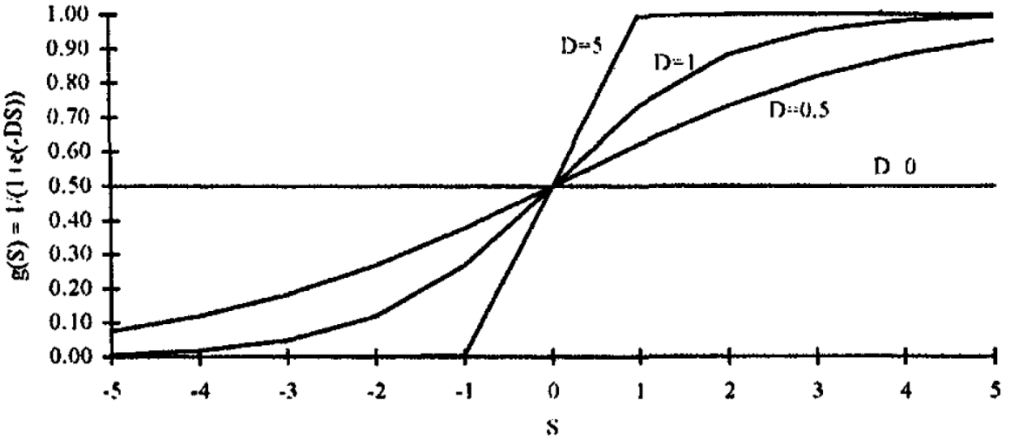

Figure 5: General form of sigmoidal functions (Lowe & Pryor 1996)

4.2.3.3 Choice of Network Architectures and Hyperparameters

In the space of possible neural network architectures, there are an infinite number of models that can be trained. For the scope of this thesis, the following architectures were chosen:

Single-layer architectures: (33-40-1), (33-80-1), (33-100-1), (33-120-1)

Double-layer architectures: (33-80-40-1), (33-100-60-1), (33-120-60-1)

For each model, the baseline value of learning rate was set to 0.05, and that of batch size was set to 8000. A 5-fold cross-validation was performed in order to determine its predictive power. The CV MSE's for the above tested models are shown below.

Model	**MSE**
33-40-1	1.8×10^6
33-80-1	1.68×10^6
33-100-1	1.65×10^6
33-120-1	1.64×10^6
33-80-40-1	1.72×10^6
33-100-60-1	1.67×10^6
33-120-60-1	1.78×10^5

Table 5: 5-fold CV Results on Baseline Model Architectures

In general, two conclusions can be made. First, we see that, in general, networks with two hidden layers perform better than those with one hidden layer. Second, for both single-layer and double-layer networks, the more number of nodes there are in the first hidden layer, the better the model performance. Given the baseline values of the hyperparameters outlined above, the (33-120-60-1) model performs best. Hence, this was chosen for further tuning of hyperparameters. For learning rate and batch size, the following values were considered.

Learning Rate: 0.01, 0.05, 0.1

Batch size: 3000, 8000, 10000

For every value of batch size, 3 models were tested, one for every value of learning rate considered. Thus, for this step, 9 models were tests. A 5-fold CV was performed on each model in order to determine and compare predictive accuracy.

Batch Size	**Learning Rate**	**MSE**
3000	0.01	3.61×10^5
3000	0.05	7.1×10^4
3000	0.1	3.38×10^4
8000	0.01	9.67×10^5
8000	0.05	1.78×10^5
8000	0.1	8.96×10^4
10000	0.01	1.26×10^6

10000	0.05	2.43×10^5
10000	0.1	1.17×10^5

Table 6: 5-fold CV Results for ANN Hyperparameter Tuning

The CV error was lowest for the model with batch size 3000 and learning rate 0.1. Thus, the best neural network model was found to be a (33-120-60-1) model with batch size 3000 and learning rate 0.1.

5. Model Comparison

5.1 Test MSE

Each of the 3 models was fitted on the test dataset, and the MSE's were found. The results are summarized below.

Model	**Test MSE**
Poisson-Gamma GLM	2.21×10^6
Tweedie GLM	2.22×10^6
Neural Network	2.23×10^6

Table 7: Test MSE's

The above results show that, in terms of precision, the Poisson-Gamma GLM performs better than the Tweedie GLM, which in turn performs better than the Neural Network model.

A plot of the residuals against the fitted values is shown below for all three models.

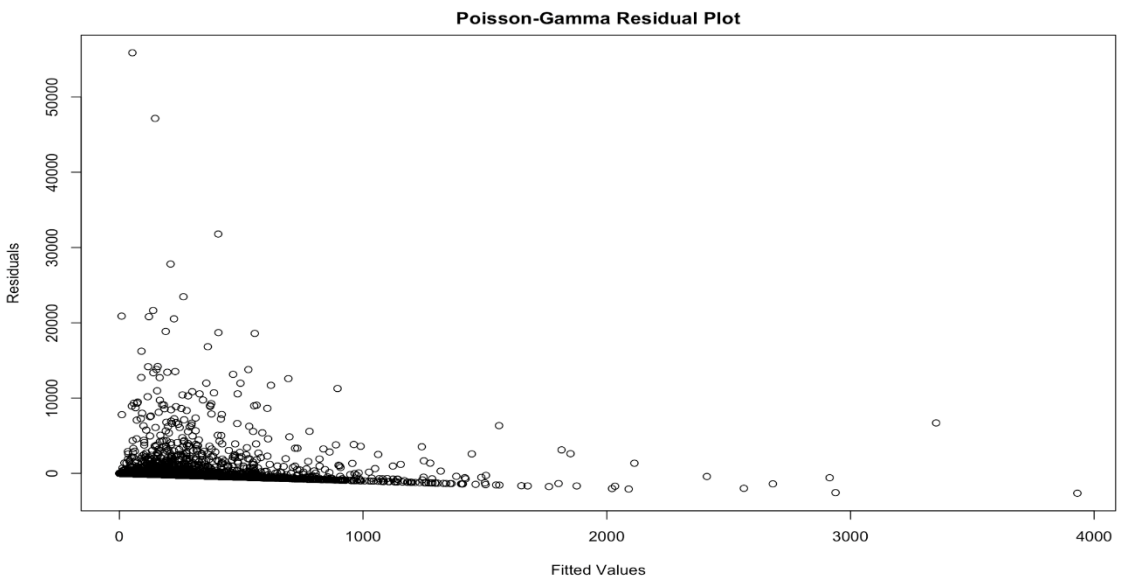

Figure 6: Poisson-Gamma Residual Plot

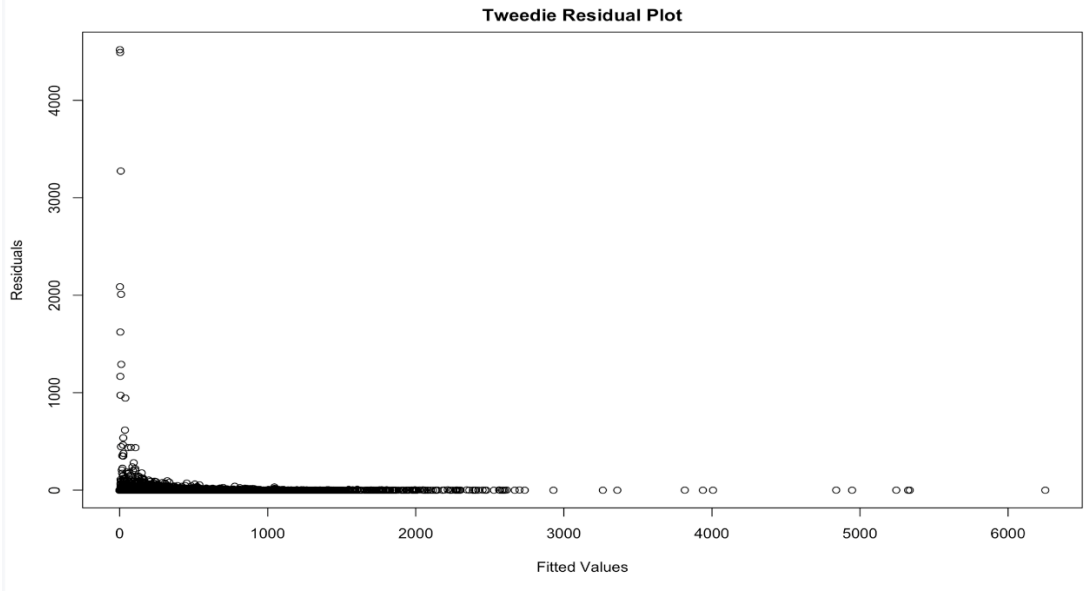

Figure 7: Tweedie GLM Residual Plot

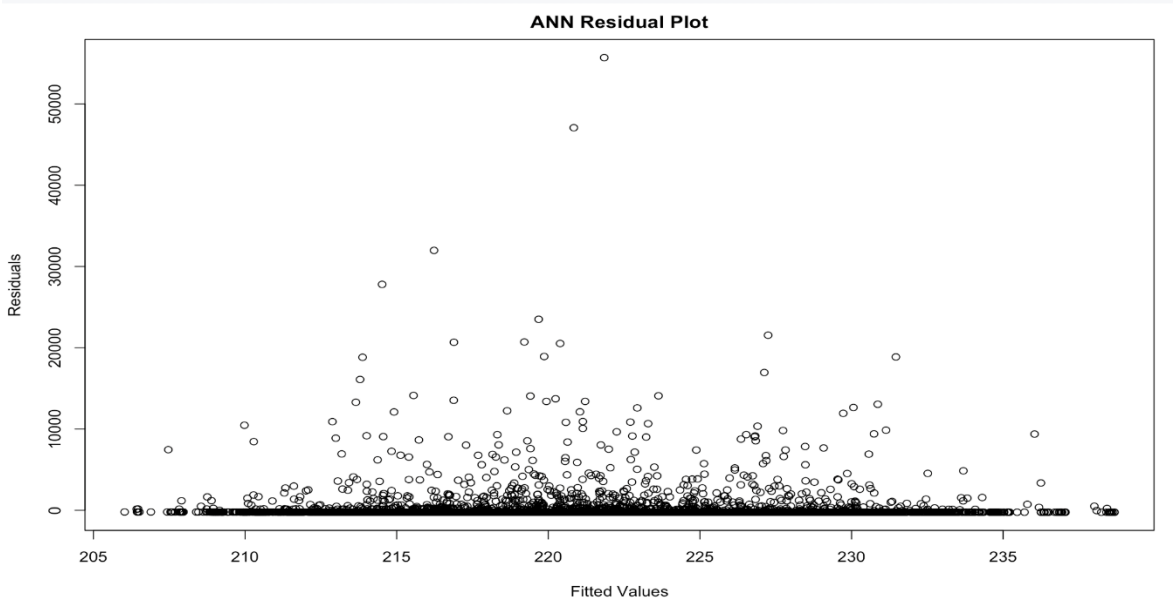

Figure 8: Neural Network Residual Plot

The residual plots seem to indicate that the Tweedie GLM is a better fit than the Poisson-Gamma GLM on the test dataset. There is clearly much less dispersion in the residuals in the Tweedie plot, where the residuals are a lot more clustered around 0 for different predicted values of risk premium. Further, the Poisson-Gamma residuals show slightly more dispersion than the ANN residuals. As a result, just by looking at the test MSE's, it is hard to accurate compare the predictive power of these models.

5.2 5-fold Cross Validation

A 5-fold cross validation was performed on all 3 models using the entire dataset. This was done to generate a better picture of the overall predictive power of each model than the test MSE and residual plots. A 5-fold cross-validation was chosen in order to minimize variance-bias trade-off. While a 10-fold cross-validation would also achieve this goal, this would

computationally be more time-consuming. The cross-validation MSE's are given the table below.

Model	CV MSE
Poisson-Gamma GLM	1.69×10^6
Tweedie GLM	1.71×10^6
ANN	1.71×10^6

Table 8: 5-fold CV MSE's

The above results show that a Poisson-Gamma GLM seems to perform slightly better than both the Tweedie GLM and the ANN. For all models, the CV MSE's are lower than the test data MSE's. The cross-validation confirms the initial conclusion that, in terms of predictive accuracy, the Poisson-Gamma GLM is slightly better than the other models. On the other hand, while the test MSE's show that the Tweedie GLM performs better than the Neural Network, the cross-validation shows that both models are equally accurate.

5.3 AIC

The AIC's for the Poisson-Gamma model, Tweedie GLM and the most optimal Neural Network are summarized below.

Model	AIC
Poisson-Gamma GLM	1.32×10^5

Tweedie GLM	1.32×10^5
ANN	1.32×10^5

Table 9: AIC Values

The AIC's are almost the same for all 3 models. This indicates that the all models can relatively equally approximate the distance between the predicted trends and the actual trends in claim costs. In other words, the AIC does not seem to present any distinctions between the models in terms of goodness-of-fit. Further analysis is, thus, needed.

5.4 Risk Premium Ratios

The aggregated risk premium ratios for all models are given in the table below. The aggregated risk premium ratio was calculated as the ratio of the total claim cost to the total expected risk premium in the entire test dataset.

Poisson-Gamma GLM	**Tweedie GLM**	**ANN**
1.114	0.917	0.893

Table 10: Aggregated Risk Ratios

This ratio is greater than 1 for the Poisson-Gamma model, and less than 1 for the Tweedie GLM and the ANN. Hence, from an actuarial standpoint, if only the expected risk premium is considered, the Tweedie GLM and the ANN are the adequate and fair models. From a profitability standpoint, the ANN seems to be the best model, since it has the lowest risk

premium ratio. In other words, given the expected risk premiums and claim costs, the profit margin is the highest for the ANN.

Risk premium ratios were also calculated by risk factor, i.e., for each category of all explanatory variables in the data. For every such variable, the data was grouped by its levels, and the risk ratio was found for each level. These have been summarized in the subsequent tables. All tables have been arranged in increasing order of the number of observations of each category in the test dataset.

5.4.1 Vehicle Age

Vehicle Age Group	Count	Poisson-Gamma GLM	Tweedie GLM	ANN
1	1865	1.19	0.97	0.77
2	2213	1.24	1.05	1.08
3	2421	0.96	0.8	0.81
4	2545	1.09	0.88	0.9

Table 11: Risk Premium Ratios for Vehicle Age

Both GLM's show the same trends in terms of risk division among groups for vehicle age. Group 3 seems to be the least risky, while Group 2 seems to be the riskiest. It is interesting to note that the ANN reflects a different trend. According to this model, Group 1, i.e., the newest vehicles, carry less risk than Group 3, but here too Group 2 is the riskiest, followed by Group 4 (the oldest vehicles).

5.4.2 Driver Age

Driver Age Group	Count	Poisson-Gamma GLM	Tweedie GLM	ANN
1	906	1.13	0.76	1.28
6	911	0.82	0.54	0.51
5	1462	1.21	1.03	0.64
2	1777	1.09	0.97	1.12
4	1962	1.23	1.27	0.99
3	2026	1.06	0.83	0.85

Table 12: Risk Premium Ratios for Driver Age

For driver age categories, we see some difference in trends between models. All 3 models are similar in that they identify Age Group 6, i.e., older drivers, as the lowest risk group. This is in line with real world trends, as older drivers tend to be more careful. While both Poisson-Gamma and Tweedie GLM's identify Group 4 as having the highest risk, the Neural Network identifies Group 1 as the riskiest. The ANN is closest to identifying the trend that, as drivers get older, their risk of incurring vehicle damage decreases.

5.4.3 Gender

Gender	Count	Poisson-Gamma GLM	Tweedie GLM	ANN
M	4434	1.25	0.98	0.96
F	4610	1.00	0.86	0.83

Table 13: Risk Premium Ratios for Gender

All 3 models are consistent in that they identify male drivers as having a higher risk of incurring vehicle damage than female drivers. The Poisson-Gamma GLM, however, cannot be followed here as it undercharges both groups. The trends detected by the models are in line with the real world, as women generally are better drivers than men. On the other hand, these trends might arise just because there are more female drivers than male drivers in the data.

5.4.4 Vehicle Body Type

Vehicle Body Type	Count	Poisson-Gamma GLM	Tweedie GLM	ANN
RDSTR	4	0	0	0
CONVT	8	0	0	0
BUS	10	3.02	2.3	2.26
MCARA	36	0.49	0.49	0.16
COUPE	119	1.3	1.2	1.25

MIBUS	120	1.5	1.51	0.99
PANVN	138	0.87	0.63	0.72
HDTOP	285	0.83	0.79	0.65
TRUCK	319	1.39	0.77	1.08
UTE	814	1.8	1.54	0.9
HBACK	2066	1.09	0.91	1.06
SEDAN	2525	0.84	0.68	0.75
STNWG	2600	1.32	1.13	0.9

Table 14: Risk Premium Ratios for Vehicle Body Type

For all 3 models, the risk premium ratios for roadsters and convertibles are negligible, because of the lack of observations of this type in the data as well as claim costs being negligible for these groups. Across all models, buses are found to be the riskiest category. This makes sense, since buses are more at risk of high claims in case of damage, which in turn is more likely to occur given the general nature of the usage of buses as well as their size. For the groups with significant claims and exposure, caravans or motorhomes are least at risk. A possible explanation for this is that they are less likely used than other types of vehicles. In interesting observation and a possible anomaly is that, while both GLM's classify larger vehicles such as trucks, minibuses and utility vehicle as having higher risk than smaller vehicles, the ANN places coupes as having the highest risk after buses.

5.4.5 Area

Area	Count	Poisson-Gamma GLM	Tweedie GLM	ANN
F	609	1.04	0.81	1.04
E	993	1.81	1.37	1.12
D	1317	1.05	0.7	0.57
B	1794	0.87	0.71	0.73
A	2002	1.28	1.21	0.97
C	2329	1.01	0.86	1.00

Table 15: Risk Premium Ratios for Vehicle Body Type

Looking at the risk ratios, it is apparent that Area D is the least risky area, while Area E is the riskiest. A possible interpretation is that Area D is the safest, while Area E is the last safe in terms of vehicle related crimes, such as car thefts or break-ins. Area B is also a relatively low risk area.

5.4.6 Vehicle Value (Measured in $10,000's)

Vehicle Value Group	Count	Poisson-Gamma GLM	Tweedie GLM	ANN
15+	4	0	0	0
10 – 15	9	1.7	1.65	1.00

5 – 10	220	1.06	0.96	0.62
0 - 5	8811	1.12	0.92	0.91

Table 16: Risk Premium Ratios for Vehicle Value

All models classify vehicles costing $15,000 or more as having the lowest risk. While this is not expected, a possible explanation is that there are only 4 observations of this category in the test dataset. For all other categories, the Tweedie GLM places the lowest risk on vehicles costing between $0 and $5000, with risk increasing as vehicles become more expensive. This is in line with the expected trend, as more expensive vehicles should be more at risk of damage. On the other hand, the Poisson-Gamma GLM and the Neural Network place a lower risk on vehicles costing between $5,000 and $10,000 than on vehicles costing between $0 and $5,000. A possible explanation for this is that the cheapest vehicles might not be in a very good condition (for instance, they may be used and older vehicles), thereby leading to a higher risk of damage and claims. Overall, excluding the most expensive vehicles (Group 15+), risk of damage increases as vehicles become more expensive.

6. Conclusions

This thesis explored 3 statistical methods for pricing insurance. These covered a broad range of spectra in terms of learning autonomy, i.e., how much of the distribution of the data is known and specified prior to modeling. On one end was the Poisson-Gamma models for claim frequency and claim severity, where the distributions of the response variables were specified. In the Tweedie

GLM, the model was allowed to deduce the distribution of the response variable under the constraint that the variance power parameter p was between 1 and 2, indicating that the data followed a compound Poisson distribution. In the case of the Neural Network, nothing is specified or assumed about the distribution. The model is allowed to determine the learning weights and optimize them without any restrictions placed on the possible distribution of claims.

In terms of predictive power on the test dataset, the Poisson-Gamma GLM marginally seems to be the best fit. A 5-fold cross validation on the entire dataset showed the Poisson-Gamma model performs marginally better than the Tweedie GLM and the Neural Network. A plot of the residuals vs. fitted values in the test data showed that the Tweedie GLM might be a better fit than the Poisson-Gamma model. Comparing the AIC's of the models, it was evident that all models are equally good fits on the test dataset.

Combining these results with an analysis of risk premium ratios on the aggregated test dataset showed that, while the Poisson-Gamma model might have a low test MSE, it is not an actuarially fair model. The Tweedie GLM and the Neural Network were actuarially fair in that, in total, they did not overcharge consumers. Breaking this analysis down for each risk factor rendered conclusions on which groups were more responsible for increasing the risk of the insured pool. The groups that were clearly identified as having a higher risk were male drivers, younger drivers, potentially unsafe areas, larger and more heavy-duty vehicles (such as buses, utility vehicles and minibuses). Overall, the Tweedie GLM and the Neural Network were able to identify trends between different groups for each rating factor, as well as price these groups, better than the Poisson-Gamma GLM. This is evident from the fact that, for the Poisson-Gamma model, almost

all of the risk premium ratios were greater than 1, indicating that this model undercharged consumers.

Overall, the results show that more autonomous models such as Tweedie GLM's and, particularly Neural Networks, are quite viable and implementable alternatives to the traditional Poisson-Gamma GLM. In terms of predictive power, Neural Networks are comparable to both Poisson-Gamma models and Tweedie models. From a business standpoint too, they can yield rates that are profitable. Further, they are easy to train since they do not require any prior knowledge of distributions and patterns in the data.

There are some potential drawbacks to using Neural Networks as opposed to Tweedie GLM's. The most obvious one is that Neural Networks can take more time to run than Tweedie models. This can be solved by using better computing systems such as GPU's, and research on how to improve the speed of deep learning is being actively pursued by firms such as Google. Another drawback of Neural Networks stems from an advantage that it has over most models. While an ANN can detect hidden patterns in the data better than most statistical learning algorithms without any external input on the same, the results are less interpretable as compared to those from, say, a Tweedie GLM, even though they may be better or more accurate. The use of ANN's can, therefore, lead to a trade-off between accuracy and interpretability. This could, however, be solved given more time, computing power and research, since Neural Networks clearly have a lot of potential in the insurance industry.

There is also scope for further research into using other machine learning algorithms for pricing insurance. Models such as k-means clustering, Support Vector Machines (SVM's), regression trees and Random Forests can be further explored, as these have previously shown some potential. Moreover, since GLM's have practical advantages than Neural Networks, especially in terms of interpretability, more research could be done on ways to combine the two models. A potential area of focus could be developing a method to incorporate offsets into neural networks, so that exposure could be better used in the modeling process.

References

Fox, J., *Applied Regression Analysis & Generalized Linear Models (Third Edition)*, Sage Publications, 2016

Andersen, D. et al, *A Practitioner's Guide to Generalized Linear Models: A foundation for theory, interpretation and application*, in CAS 2004 Discussion Paper Program

Smyth, G.K., Jørgensen, B., *Fitting Tweedie's Compound Poisson Model to Insurance Claims Data: Dispersion Modelling*, in ASTIN Bulletin, Vol. 32, No. 1, 2002, pp. 143 – 157

Yan, J., Guszcza, J., Flynn, M., Wu, C.P., *Applications of the Offset in Property-Casualty Predictive Modeling*, in Casualty Actuarial Society *E-Forum*, Winter 2009

Kuhn, M., Johnson, K., *Applied Predictive Modeling*, Springer, 2016

Rumelhart, D. E., Hinton, G. E., Williams, R. J., *Learning representations by back-propagating errors*, in Nature Vol. 323, October 1986

Hastie, T., Tibshirani, R., Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference and Prediction (Second Edition)*, Springer, 2009

James, G. et al, *An Introduction to Statistical Learning with Applications in R*, Springer, 2013

Cybenko, G., *Approximation by Superpositions of a Sigmoidal Function*, in Mathematics, Controls, Signals and Systems (1989) 2: 303 – 314

Lowe, J., Pryor, L., *Neural Networks v. GLM's in pricing general insurance*, presented at 1996 General Insurance Convention

Burnham, K. P., Andersen, D. R., *Model Selection and Multimodel Inference: A Practical Information – Theoretic Approach (Second Edition)*, Springer, 2002

Panchal, G. et al, *Searching Most Efficient Neural Network Architecture Using Akaike's Information Criterion (AIC)*, 2010 International Journal of Computer Applications (0975 – 8887), Volume 1 – No. 5