PREDICTIVE MODELLING OF AUTO INSURANCE CLAIMS

Ernest Tamekloe (eetamek@ilstu.edu)

Kelvin Nii Lartey-Abrahams (knlarte@ilstu.edu)

04/01/2021

Abstract

Calculating actuarial premium is a key function of insurance companies. One main component of pricing auto insurance is loss cost, also known as pure premium or pure cost, which is the amount of money an insurer must pay to cover claims, including the costs to administer and investigate such claims. This work focuses on predictive modelling of loss costs for the auto insurance line of business. Two main approaches to loss cost modelling here are discussed here. First is fitting separate models for claim frequency (using Poisson and Negative Binomial distributions) and claim severity (using Gamma and Lognormal distributions). The second is fitting one model for pure premium using the Tweedie distribution. Exploratory data analysis, construction of the models, evaluation and results comparison, challenges and conclusion are all discussed. Since this study was limited to the auto-insurance line of business, suggestions were made to extend the analysis to other porperty and casualty lines of business as well as making use of real world data.

Contents

1. Introduction

1.1 Background

Premium pricing is an important and a challenging task in insurance. Pricing involves finding the fair premium that covers an insurer's expected costs, expenses and providing a fair return to the insurer's investors. The frequency and severity of insurance claims play a major role in the pricing of the premiums. Frequency is the average number of claims per period, usually per year. Severity is the amount paid due to a loss or the (average) size of loss due to an event. It is important for insurers to adequately model past and current data on claim experience to be able to settle claims from existing or future portfolios and use these models to project the expected future experience in claim amounts. This will in turn provide leverage for actuaries to adequately price insurance products.

Predictive modelling of auto insurance claims is gaining grounds in the insurance and consulting space due to the capabilities and leverage it provides. The Society of Actuaries is gradually seeing the immense importance of predictive modelling in the work of the actuary and this has led to the introduction of Statistics for Risk Modelling (SRM) and Predictive Analytics (PA) exams into the current exam syllabus while the Casualty Actuarial Society also has Modern Actuarial Statistics I & II in their exams to give actuaries some predictive edge in the ever-evolving insurance space. In a paper released by the SOA titled "Considerations for Predictive Modelling in Insurance Applications", the SOA acknowledged that in a world where data and analytics are quickly taking over many industries, the Predictive Analytics and Futurism and the Modelling sections of the Society of Actuaries (SOA) along with other SOA sections, are interested in educating actuaries on how best to incorporate predictive modelling into relevant areas of actuarial practice. This underscores the importance of predictive modelling in the work of the actuary, irrespective of the area of practice.

Pure premium or loss cost modelling is one of such key areas where actuaries pay much attention to because getting the pure premium estimates right implies being able

to price insurance products correctly. Thus, actuaries do employ predictive modelling techniques to make reasonable estimates for loss costs or claim amounts and then factor this into pricing insurance products. Various approaches have been used in modelling claims data, from linear regressions to non-linear Generalized Linear Models (GLM) and machine learning approaches and each approach has unique features that make them desirable.

There may be drawbacks in using predictive modelling in forecasting pure premium or loss cost, however. In any case, past data may not necessarily be a good measure of how future claims will develop. The nature of datasets and outliers may render results unreliable. More so, there is no such thing as a "super predictive model", as no model is perfect. The famous British Statistician, George Box, summed it up in the following quote: "All models are wrong, but some are useful". Thus, various methods must be tried and tested to pick the one with the least predictive error.

Despite these shortcomings, predictive modelling provides a good go-to tool for making good guesses about the future, which forms the basis of this work.

1.2 Problem Statement & Objectives

The aim of this project is to address the problems faced by insurance companies in modelling loss cost/pure premium. Loss cost is the amount of money an insurer must pay to cover claims, including defense cost and containment (legal expenses). The study will attempt to provide a framework for choosing the appropriate statistical distribution and fitting it to the claims data in order to predict the pure premium. The best models for frequency and severity will be chosen based on goodness of fit statistics. Finally, the study will attempt to evaluate the prediction accuracy of the best-chosen models and then recommendations will be made on the overall best model for actuaries in property and casualty insurance companies.

The objectives of this study are to are to:

1. develop separate models for frequency and severity of claims data (Frequency-

Severity method).

2. develop a single model for pure premium (i.e., average claim cost involving frequency and severity).
3. Perform a comparative analysis of the best Frequency-Severity method and the best Pure Premium method.
4. evaluate the overall best method and make recommendations for actuaries.

1.3 Literature Review

Banerjee and Dasgupta (2011) defined Predictive modeling as a process for transforming data insights into an estimation of future outcomes upon which actionable decisions can be made. It is worthy to note that most companies are now making use of predictive modelling to make informed decisions and increase productivity and profitability. Auto insurance companies are beneficiaries of the leverage that predictive modelling provides in making reasonable guesses of the claims associated written policies.

Bill Lentz (2018) in his paper "The Expanding Use of Predictive Analytics in Claims Management" stipulated that with the predictive modeling process, it is not too hard to see how an extensive review of historical data can provide insights on trends. Predictive modeling compares factors associated with new and pending claims against those of past losses. He concluded that predictive modeling could provide useful insight in formulating reserves and settlement values for current losses and can also be used to identify claims with the potential for high defense costs while also identifying which defense firms were associated with favorable outcomes involving similar cases.

Andy Yohn (2021) also identified three main ways that predictive modelling can help insurance companies analyze their claims data and take strategic steps to meet their goals: triaging claims, identifying outlier claims, and transforming the claims process. He made some critical observations and concluded that predictive modelling can contribute to tighter management of budgets by employing forecasted data regarding claims, giving insurers a strategic advantage. Also, he observed that with proper mod-

elling tools, P&C insurers can review previous claims for similarities and give advance notice of potential losses or related complications that can help insurers cut down on these outlier claims and can also help insurance companies to use lessons learned from outlier claim data preemptively to create plans for handling similar claims in the future.

A more detailed work on predictive modelling of auto insurance claims was done by Yunos et. al. (2016) in which they applied Back Propagation Neural Networks (BPNN) to analyzing the Malaysian motor insurance claims data. In the paper, they highlighted two important issues in the motor insurance about claims data and techniques. The first issue is related to the characteristics of insurance data which contain massive information or large number of variables, uncertainty, information that is very noisy and incomplete information. Another problem is the existing of extreme values in the data which cannot be ignored or dealt with as outliers. In conclusion, they stated that BPNN model was successful in predictive modelling of Malaysian motor insurance claims by using several of network structures. They identified that the main advantage of using BPNN is that the model can deal with non-linear data.

Dang (2018) did work more relatable to what was done in this study in the paper "Econometrics of Insurance". The only setback to his work was in regards to the data that was used. Dang employed the GLM approach to separately analyze the frequency and severity data as well as a joint approach using the Tweedie model. The data comprised of claims data and policy information from two unrelated years. In conclusion, the Negative Binomial distribution was preferred to Poisson for the frequency models and Gamma model was preferred to the Lognormal model for the severity models. However, this result cannot be reliable because the input data from two unrelated years do not give a realistic estimates for the future. This shortcoming is addressed in our work by using claims and policy information data that relate to the same year so to fit a more realistic model for prediction.

2. Data & Methodology

The data used for this project is taken from the Third Actuarial Pricing Game, as part of a research conducted by Arthur Charpentier, Universite De Rennes I (France) & Quantact (Montreal, Canada) with the support of the French Institute of Actuaries. There are two datasets each including 100,000 insureds for Year 0:

1. An underwriting dataset with information about insurance policies, insured drivers and their cars.
2. A claims dataset with all claims collected during year 0 to all policyholders, with the variables to be modeled being number of claims (claims_nb) and claim amount (claim_amount). The claim_nb variable takes a value 1 for each claim and the individual claim amounts for the claim_amount variable range from [1] $-2,000$ to $300,000$.

2.1 Data Characteristics

The variables in the data can be grouped into five classes: control variables, driver characteristics, policy characteristics, vehicle characteristics and response variables as shown in Table 1.

Below is a short description of each variable:

1. **id_client**. **id_client** is a string of the form Annnnnnnn ('A' followed by an 8-digit number). First client ID is A00000001 and last is A00091488.
2. **id_vehicle**. **id_vehicle** is a string of the form Vnn (a 'V' followed by a 2-digit number). First vehicle is always numbered V01. If a client has multiple vehicles, then the numeration increases by 1.
3. **id_policy**. **id_policy** is a string of the form Annnnnnnn-Vnn, resulting from the concatenation of **id_client** and **id_vehicle**.

[1] negative claims occur when the provision made for outstanding claims is unduly high. When the claim amount, which is lower than the amount of outstanding provision is paid, there will be a negative incurred claim amount.

Table 1: Available Variables In Dataset

Control	Driver	Policy	Vehicle	Response
id_client	drv_age_lic1	pol_bonus	vh_age	claim_amount
id_policy	drv_age_lic2	pol_coverage	vh_cyl	claim_nb
id_vehicle	drv_age1	pol_duration	vh_din	
id_year	drv_age2	pol_sit_duration	vh_fuel	
	drv_drv2	pol_insee_code	vh_make	
	drv_sex1	pol_pay_freq	vh_model	
	drv_sex2	pol_payd	vh_sale_begin	
		pol_usage	vh_sale_end	
			vh_speed	
			vh_type	
			vh_value	
			vh_weight	

4. **id_year**. Year ID begins at Year 0 and ends at Year 4.

5. **pol_bonus**. Represents the bonus/malus system (a system offering a financial incentive, or bonus, for the purchase of cars with low carbon emissions, and a fee, or malus, for the purchase of high-emission vehicles) but is used here as a possible feature. The coefficient is attached to the driver. It starts at 1 for young drivers (i.e. first year of insurance). Then, every year without claim, the bonus decreases by 5% until it reaches its minimum of 0.5.

6. **pol_coverage**. There are 4 types of coverage: Mini, Median1, Median2 and Maxi, in this order. Mini policies cover only Third Party Liability claims, whereas Maxi policies covers all claims including damage, theft, windshield breaking, Assistance, etc.

7. **pol_duration**. Policy duration represents how old the policy is. It is expressed in years,

8. **pol_sit_duration**. Situation duration represent how old the current policy characteristics are. It could be different from **pol_duration**, because the same insurance policy could have evolved in the past (e.g. by changing coverage, or

vehicle, or drivers).

9. **pol_ pay_freq**. The price of the insurance coverage can be paid annually, bi-annually, quarterly or monthly.

10. **pol_payd**. **pol_payd** is Boolean (i.e. a string with Yes or No), which indicates whether our client has subscribed a mileage-based policy or not.

11. **pol_usage**. The policy use describes how the driver uses his vehicle. There are 4 possible values: **Work Private** which is the most common, **Retired** which is presumed to be aimed at retired people who also are presumably driving less kilometers, **Professional** which denotes a professional usage of the vehicle, and **All Trips** which is quite similar to Professional (including pro tours).

12. **pol_insee_code**. This is a 5-digits alphanumeric code used by the French National Institute for Statistics and Economic Studies (hence INSEE) to identify communes and departments in France.

13. **drv_drv2**. The **drv_drv2**. Boolean (Yes/No) identifies the presence of a secondary driver on the vehicle. There is always a first driver, whose characteristics (age, sex, licence) are provided, but a secondary driver is optional, and is present 1 time out of 3.

14. **drv_agel**. This is the age of the first driver. **drv_age1** is expressed in years counted from the beginning of the considered year. Then, **drv_age** increases by 1 every year, like in real world.

15. **drv_age2**. When **drv_drv2** is Yes, then the secondary driver's age is present. When not, this age is 0.

16. **drv_sexl**. Same price for women and men by insurers in Europe. But driver's gender can still be used in academic studies, and that's why **drv_sexl** is available in the datasets, and can be used as discriminatory variable in this pricing game.

17. **drv_sex2**. As for **drv_sex1**, **drv_sex2** represents the gender of the optional secondary driver.

18. **drv__age__licl**. **drv__age__licl** is the age of the first driver's driving licence.

19. **drv__age__lic2**. **drv__age__lic2** is the age of the second driver's driving licence.

20. **vh__age**. This variable is the vehicle's age, the difference between the year of release and the current year.

21. **vh__cyl**. The engine cylinder displacement is expressed in ml in a continuous scale. This variable should be highly correlated with din power of the vehicle.

22. **vh__din**. The **vh__din** is a representation of the motor power.

23. **vh__fuel**. **vh__fuel** has mainly two values: Diesel and Gasoline. Very few Hybrid vehicles can also be found.

24. **vh__make**. The make (brand) of the vehicle. As the database is built from a french insurance, the three major brands are Renault, Peugeot and Citroën.

25. **vh__model**. As a subdivision of the make, vehicle is identified by its model name. The are about 100 different make names in the datasets, and about 1,000 different models.

26. **vh__sale__begin** and **vh__sale__end**. **vh-_sale___begin** and **vh__sale__ end** are the ages from the beginning of the current year and the end of marketing years of the vehicle. This could for instance identify policies that cover very new vehicles or second-hand ones.

27. **vh__speed**. This is the maximum speed of the vehicle, as stated by the manufacturer.

28. **vh__type**. **vh__type** can be Tourism or Commercial. You'll find more Commercial types for **Professional** policy usage than for **Work Private**.

29. **vh__value**. The vehicle's value (replacement value) is expressed in euros, without inflation so it should be stable from a year to another.

30. **vh__weight**. **vh__weight** is the weight (in kg) of the vehicle.

31. **id__claim**. **id__claim** is a string of the form CLnn (CL followed by a 2-digit number). Numbering of the claims begins at 1 for every policy and each year.

Then, the last value of **id_claim** is the maximum number of claims for a vehicle in a year.

32. **claim_nb**. As we are talking about individual claims, each **claim_nb** has a value of 1.

33. **claim_amount**. Individual claim amounts.

2.2 Modifications To the Data

1. In the claims dataset, each client might make several claims with the same vehicle, which leads to duplication in id_client $*$ id_vehicle. Since this will be a problem when merging the Underwriting and Claims datasets, we summed up all the times and claim amounts for each occurrence of id_client $*$ id_vehicle to prevent duplicates.

2. The merged dataset, which was used in conducting analysis, has $100,000$ observations with 33 variables.

3. The merged dataset was split into 70%-train for building the models, and 30%-test for predictive purposes.

2.3. Methodology

This subsection discusses the framework for GLM which is key to understanding how to model frequency and severity of claims. GLMs are preferred to linear models because of the following reasons:

1. They have relaxed assumptions such that the response variable y is linked to a linear function of predictor variables x_i with a nonlinear link function.
2. The variance in the response variable y does not have to be constant across observations but can be a function of y_i's expected value.

So with GLMs, the response variable y can have a distribution in the linear exponential family, which includes distributions important to actuaries: Poisson, binomial, normal, gamma, inverse-Gaussian, and compound Poisson-Gamma. With these distributions, actuaries can model frequency, severity, and loss ratios. The likelihood function has a key role in GLMs. Maximum likelihood estimation replaces least squares in the estimation of model coefficients. The log-likelihood function, Akaike Information Criterion(AIC) and Mean Square Error(MSE) are used to perform statistical tests.

2.3.1 Problems With Linear Models

Under multiple linear regression models, the response variables y_i , with $i = 1, \ldots, n$, acts as a linear function of predictor variables x_{ij} , plus a constant β_0 such that

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_k x_{ik} + \epsilon_i \tag{1}$$

Two key assumptions are that error terms ϵ_i have expected value 0, and that the variance of ϵ_i is constant and does not change across observations i: $Var(\epsilon_i) = \sigma^2$. The errors ϵ_i are usually assumed to be independent and normally distributed. The coefficients in the linear model are estimated by least squares estimation. Linear models have shown their value in modelling, but the following example demonstrates the

problems that can arise when using the linear models:

1. The Poisson distribution is commonly used to model the number of claims. If y_i is Poisson distributed, then $Var(y_i) = E(y_i)$. Here, the variance is not constant across observations, but depends on the expected value of the response variable. So, the assumption of constancy of the variance across risks, $Var(y_i) = Var(\epsilon_i) = \sigma^2$, is invalid.

2. The left hand side of equation (1) needs to be non-negative when modeling the expected number of claims but this cannot be guaranteed in the linear model because there is a possibility that some combination of the predictors x_{ij} could result in a negative value.

3. Rather than building an additive model where the contributions of risk characteristics $x_{i1}, x_{i2}, \ldots, x_{ik}$ are added, perhaps a multiplicative model is more appropriate.

This implies that the linear model needs to be adjusted. Let $\ln(E[y_i])$ equal the linear combination of predictor variables which addresses issues 2 and 3. Equation (1) becomes $\ln(E[y_i]) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}$ giving

$$E[y_i] = e^{\beta_0} e^{\beta_1 x_{i1}} \ldots e^{\beta_k x_{ik}}$$

.

In this model, the expected number of claims for risk i, $E[y_i]$, will not be negative, and the predictive model is multiplicative but this adjustment raises another issue: how are the coefficients β_j in this nonlinear equation determined?

2.3.2 GLM Assumptions

GLMs generalize linear models in the following two important ways:

1. The independent response variables y_i can be connected to a linear function of predictor variables with a nonlinear link function.

2. The variance in the response variables y_i does not need to be constant across risks, but can be a function of y_i's expected value.

3. Random variables y_i can be members of a linear exponential family of distributions.

Elaborating on the third point, a modeler can choose a distribution from the linear exponential family that is appropriate for application. Choosing a particular distribution for the model means that maximum likelihood estimation can be used to calculate the coefficients, and algorithms exist for computing the coefficients that work for all distributions in the exponential family.

The GLM equation for response random variables y_i is

$$g(E[y_i]) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}$$

The link function $g()$ can be a nonlinear function but there are restrictions on it: it should be differentiable and strictly monotonic. The inverse function of the strictly monotonic function exists and this can be rewritten from the previous equation as

$$E[y_i] = g^{-1}(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}) \tag{2}$$

2.3.3 Exponential Family

Distributions used by actuaries share a common structure and can be grouped into an exponential family. For GLMs, the response variable y is assumed to have a probability distribution function that can be written as

$$f(y; \theta, \phi) = \exp\left[\frac{y\theta - b\theta}{a(\phi)} + c(y, \phi)\right]$$

Parameter θ is often referred to as the canonical parameter or parameter of interest. Parameter ϕ is called the dispersion parameter or the nuisance parameter because the

distribution's mean does not depend directly on ϕ. The functions $b(\theta)$, $a(\phi)$, and $c(y, \phi)$ determine the type of distribution. The mean and variance of the distribution are

$$E[y] = b'(\theta) \tag{3}$$

$$Var[y] = a(\phi)b''(\theta), \tag{4}$$

Table 2 displays the distributions used in this paper in the exponential family form.

Table 2: Exponential Family Form

Common form of pdf	θ	$b(\theta)$	ϕ	$a(\phi)$	$c(y, \phi)$
Poisson: $\lambda^y e^{-\lambda}/y!$	$\ln \lambda$	e^θ	1	1	$-\ln(y!)$
Negative Binomial: $\binom{x_i-1}{k-1} p_i^k (1-p_i)^{x_i}$	$\log(1-p_i)$	$-k \log[(1-e^{\theta_i})/e^{\theta_i}]$	1	1	$-\log \binom{y_i-1}{k-1}$
Gamma: $\beta^\alpha y^{\alpha-1} e^{\beta y}/\Gamma(\alpha)$	$-\frac{\beta}{\alpha}$	$-\ln(-\theta)$	$\frac{1}{\alpha}$	ϕ	$\frac{1}{\phi} \ln \frac{y}{\phi} - \ln y - \Gamma(\frac{1}{\phi})$
Normal: $\frac{1}{\sqrt{2\pi\sigma^2}} \exp[-\frac{(y-\mu)^2}{2\sigma^2}]$	μ	$\theta^2/2$	σ^2	ϕ	$-\frac{1}{2}[\frac{y^2}{\phi} + \ln(2\pi\phi)]$

2.3.4 The Variance Function and the Relationship between Variances and Means

If $a(\phi)$ is replaced with ϕ/w_i, the variance formula from (4) becomes

$$Var[y_i] = \frac{\phi}{w_i} b''(\theta_i) \tag{5}$$

When we invert formula (3) for the mean

$$\mu_i = b'(\theta_i)$$

$$\theta_i = b'^{-1}(\mu_i)$$

,

and we replace θ_i in (5) by the right hand side above gives $Var[y_i] = \frac{\phi}{w_i} V(\mu_i)$

$V(\mu_i)$ is known as the variance function and it defines the relationship between the variance and the mean for a distribution in the exponential family. Table 3 shows the variance function for certain distributions used in this paper.

Table 3: Exponential Family Form

Distribution	$V(\mu)$	Distribution	$V(\mu)$
Normal	$\mu^0 = 1$	Tweedie	$\mu^p, 1 < p < 2$
Poisson	μ	Gamma	μ^2

2.3.5 Maximum Likelihood Estimation

GLM coefficients are estimated using Maximum Likelihood Estimation (MLE).The likelihood function is

$$L(y;\beta) = \prod_{i=1}^{n} \exp\left[\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c_i(y_i,\phi)\right]$$

However, It is easier to maximize the log-likelihood:

$$l(y;\beta) = \sum_{i=1}^{n} \left[\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c_i(y_i,\phi) \right]$$

To maximize the log-likelihood, the partial derivatives with respect to the $\beta_j's$ are calculated and set to zero:

$$\frac{\partial l(y;\beta)}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial}{\partial \beta_j} \left[\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c_i(y_i,\phi) \right] = \sum_{i=1}^{n} \frac{1}{a_i(\phi)} \left[y_i \frac{\partial \theta_i}{\partial \beta_j} - \frac{\partial b(\theta_i)}{\partial \beta_j} \right]$$

Recall $\mu_i = b'(\theta_i)$ and let $g(\mu_i) = \nu_i$ so that $\nu_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}$. From the chain rule,

$$\frac{\partial}{\partial \beta_j} = \frac{\partial}{\partial \theta_i} \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \nu_i} \frac{\partial \nu_i}{\partial \beta_j}$$

After applying chain rule, we have the following:

$$\frac{\partial l(y;\beta)}{\partial \beta_j} = \sum_{i=1}^{n} \frac{(y_i - \mu_i)x_{ij}}{a_i(\phi)b''(\theta_i)g'(\mu_i)}$$

Recall $Var[y_i] = a_i(\phi)b''(\theta_i)$ and can be rewritten as $Var[y_i] = (\phi/w_i)V(\mu_i)$ and this gives

$$\frac{\partial l(y;\beta)}{\partial \beta_j} = \sum_{i=1}^{n} w_i \frac{(y_i - \mu_i)x_{ij}}{(\phi/w_i)V(\mu_i)g'(\mu_i)} \tag{6}$$

From the equation, the weighted sum of the differences $(y_i - \mu_i)$ should equal 0. This is the MLE solution for the best coefficients $b_0, b_1, \ldots, b_k$ to predict $E[y_i]$.

2.3.6 Modeling Pure Premiums

Pure premium is expressed as Pure premium = losses/exposure.There are two approaches to modeling pure premiums. The first approach is to model frequency and

severity separately and then combine the results. The second approach is to model losses directly and this paper takes a look at both approaches.

2.3.7 Modelling Frequency and Severity Separately

Pure premiums can be split into frequency and severity components:

$$\text{pure premium} = \text{frequency} \times \text{severity} = \left(\frac{\text{number of losses}}{\text{exposure}}\right) \times \left(\frac{\text{losses}}{\text{number of losses}}\right)$$

The frequency and severity components are modeled separately, and then the predicted frequencies and severities can be multiplied together to produce pure premiums.

2.3.8 Modelling Losses Directly

A Tweedie distribution (compound Poisson-gamma) is often used to model aggregate losses. Suppose the number of losses n is Poisson distributed and loss amounts u_j are i.i.d. gamma distributed, then aggregate loss y can be written as $y = u_1 + u_2 + \cdots + u_n$. Hence, frequency and severity are combined into a single distribution and model. Tweedie family distributions are members of the exponential family with variance functions $V(\mu) = \mu^p$ where p can take on values in the ranges $(-\infty, 0] \cup [1, \infty)$. The Tweedie distribution has the variance function

$$V[\mu] = \mu^p$$

with $1 < p < 2$,

There are complications with using the Tweedie distribution. Relative Frequency and Severity components may vary among risks in the sense that some risks might have high frequency and low severity, whereas other risks have low frequency and high severity.

To circumvent this, the variance of response variables y_i is modeled as

$$Var[y_i] = \phi_i \mu^p$$

which allows dispersion parameter ϕ_i to vary across risks but p is kept constant.

2.3.9 Model Comparison

It is sometimes difficult to analyze goodness of fit and compare models due to the sheer variety of distributions under the GLM framework. The log-likelihood is used to fit models to data and is also useful in making model comparisons. The log-likelihood function for linear exponential family distributions is given as

$$l(y; \beta) = \sum_{i=1}^{n} \left[\frac{y_i \theta_i - b(\theta_i}{a_i(\phi)} + c_i(y_i, \phi) \right]$$

Define $l(y;\theta^M)$ as the the maximum value of the log-likelihood function for model M with its set of predictor variables. By itself, the value of $l(y;\theta^M)$ may not reveal much about model fit, but generally, the larger, the better. However, problems may arise when making a direct comparison of log-likelihood measures between models. One model including more predictive variables than another implies that there is good chance that the model with more predictors will have a larger value for $l(y;\theta^M)$. To address this problem, two other measures were used in the study:

1. AIC is an abbreviation for Akaike information criterion. It is given by

$$AIC = 2[l(y;\theta^M) + p]$$

   where p is the number of fitted parameters in the model. Smaller values for AIC indicate a better fit.

2. Mean Squared Error(MSE) represents the difference between the original and predicted values extracted by squared the average difference over the data set. It is given by

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

   . Smaller values for MSE indicate a better fit.

There are drawbacks to using the LR (likelihood ratio) test and the AIC. If the models are nested, the parameters in the larger model that are not in the smaller model are the ones being tested, with values specified implicitly by their exclusion from the smaller model. If the models are not nested, we cannot test for the parameters anymore, because both models have parameters that are not in the other model.

Like the LR test, AIC (not used for formal testing) is used to compare nested models so results may not be reliable when comparing non-nested models. Also the test MSE gives us an idea of how well a model will perform on a test set. However, the drawback of using only one test set is that the test MSE can vary greatly depending on which observations were used in the training and testing sets. In order to rectify this

and improve upon the MSE as a basis for model comparison, we performed a k-fold cross validation which involves fitting a model several times using a different training and testing set each time, then calculating the test MSE to be the average of all of the test MSE's.

The following are the steps to perform a k-fold cross validation:

1. Randomly divide a dataset into k folds of roughly equal size.
2. Choose one fold to be the holdout set. Fit the model on the remaining $k-1$ folds. Calculate the test MSE on the observations in the fold that was held out.
3. Repeat this process k times, using a different set each time as the holdout set.
4. Calculate overall test MSE to be average of k test MSEs.

For this study, the 10-fold cross validation was used used and it provides a more reliable basis for model comparison.

3. Analysis & Results

In this section, we start by exploring individual variables to gain a better understanding of the information we have available in our dataset.

3.1 Exploratory Data Analysis

The histograms of claim number and log(claim_amount) is shown in Figure 1. From Figure 1, the majority of policy holders have claim counts of less than 1. Table 4 shows the number of clients organized by claim counts. From Table 4, the proportion of clients having no claims is 87.35% which validates the results obtained from the histogram. This gives a hint that possible candidate models for fitting claim numbers are the zero-inflated Poisson and negative binomial models.The histogram for actual claim numbers is right skewed which means that most of the numbers is 0 or no claims were recorded most of the time. The log transformed claim numbers too took the same shape of right skewness. The claim amount is also right skewed from the histogram, but the log transformed claim amount is normally distributed or appears to be symmetrical. Thus, a potential model for claim severity is the log transformed model.

Table 4: Number of Claims by Claim Counts

Number of Claims	0	1	2	3	4	5	6
Observation	87346	11238	1264	134	16	1	1

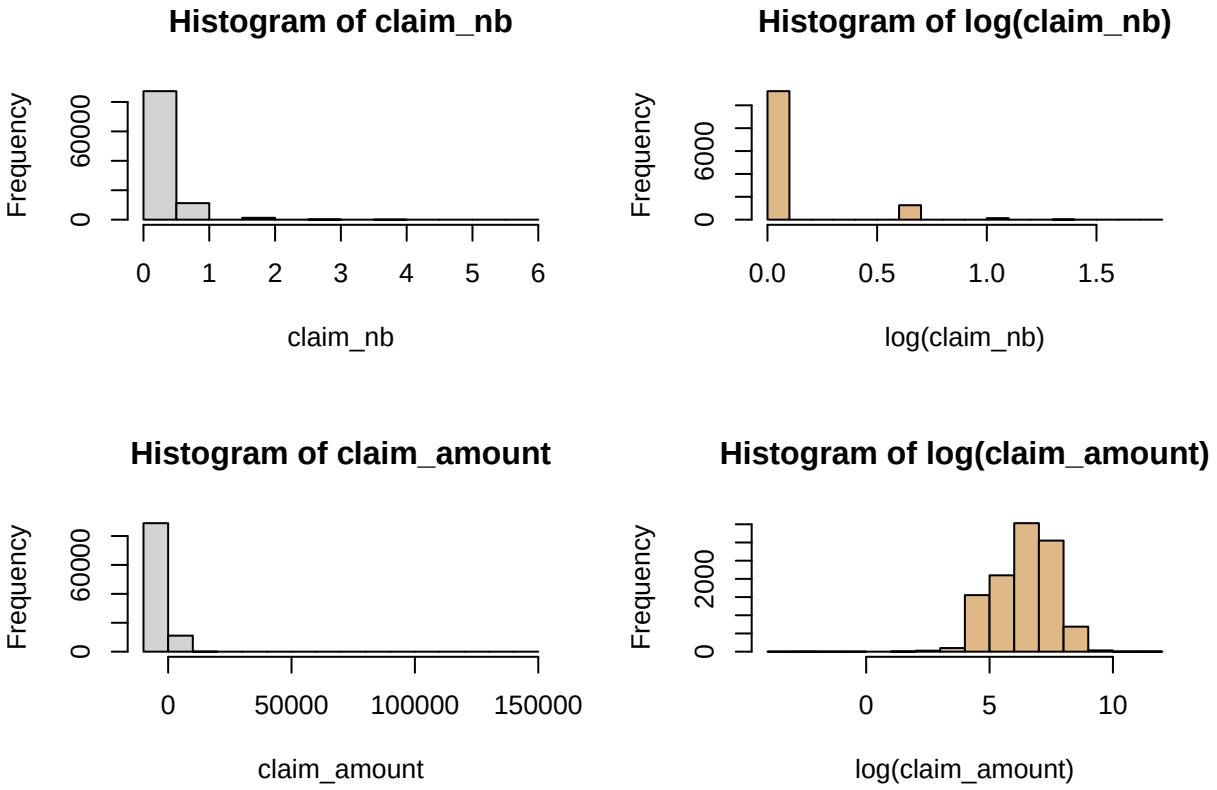

Figure 1: Histograms of Response Variables

Also, a few key summary statistics for the dataset are given below:

Table 5: Summary Statistics for Variables

Number of Observations	100,000
Number of Claims	19,243
Sum of Earned Exposures	1,108,179
Total Claim Size/Sum of loss dollars	$11,724,608
Average Claim Frequency	1.2%
Average Claim Severity	$823
Average Pure Premium	$10.6

Table 6: Summary Statistics for Variables

	Mean	SD	Median	Min	Max
id_client*	45773.2079800	26415.7934226	45785.5	1.00	91488.00
id_vehicle*	1.0921100	0.3170280	1.0	1.00	9.00
id_policy*	50000.5000000	28867.6577967	50000.5	1.00	100000.00
id_year*	1.0000000	0.0000000	1.0	1.00	1.00
pol_bonus	0.5373366	0.0980505	0.5	0.50	2.16
pol_coverage*	1.7024500	1.0375764	1.0	1.00	4.00
pol_duration	11.0817900	8.5531264	9.0	1.00	41.00
pol_sit_duration	2.7334900	2.3642881	2.0	1.00	25.00
pol_pay_freq*	2.4831100	1.2642906	2.0	1.00	4.00
pol_payd*	1.0415200	0.1994906	1.0	1.00	2.00
pol_usage*	3.5861800	0.6263680	4.0	1.00	4.00
pol_insee_code*	9963.6296800	5307.7906449	10441.0	1.00	17794.00
drv_drv2*	1.3318600	0.4708834	1.0	1.00	2.00
drv_age1	54.6838200	14.8717592	54.0	19.00	103.00
drv_age2	15.5812200	24.0037594	0.0	0.00	99.00
drv_sex1*	1.6020100	0.4894858	2.0	1.00	2.00
drv_sex2*	1.4604000	0.7109971	1.0	1.00	3.00
drv_age_lic1	32.4898500	13.4652980	33.0	1.00	111.00
drv_age_lic2	9.1252900	16.9975269	0.0	0.00	111.00
vh_age	9.5519555	7.0271179	8.0	1.00	66.00
vh_cyl	1645.8833600	461.9299742	1587.0	0.00	6997.00
vh_din	91.3924500	34.3113201	87.0	13.00	555.00
vh_fuel*	1.4520800	0.4993057	1.0	1.00	3.00
vh_make*	60.1711700	25.1221230	71.0	1.00	101.00
vh_model*	488.3030700	310.4140157	475.0	1.00	1023.00
vh_sale_begin	11.6537200	7.7897546	10.0	1.00	74.00
vh_sale_end	8.6731700	6.6437530	7.0	1.00	55.00
vh_speed	170.6830100	23.3678332	170.0	25.00	310.00
vh_type*	1.9015100	0.2979775	2.0	1.00	2.00
vh_value	18058.6914800	8663.2686598	16229.5	0.00	155498.00
vh_weight	1128.2075200	360.6409381	1130.0	0.00	7901.00
claim_nb	0.1424300	0.3973478	0.0	0.00	6.00
claim_amount	117.2460837	926.0789954	0.0	-1863.92	141828.26

3.2 Fitting Frequency-Severity Model

Using the dataset, we will construct, evaluate, and compare the following models:

1. Model F1: model claim count using a Poisson distribution
2. Model F2: model claim count using a negative binomial distribution
3. Model S1: model severity using a gamma distribution
4. Model S2: model severity using a log-normal distribution

3.2.1 Model F1

In the Poisson regression model, the dependent variable is claim_nb, which represents the number of claims per insurance policy in a specified time interval. We fit 3 Poisson models with a log-link function. The presence of an offset variable differentiates the models, as it accounts for the risk exposure of insurance policy i, which is the time interval in this case, usually expressed in years, from the initial moment when the policy was issued until the moment the sample is observed.

The following are the three models that were fit:

1. Poisson model without offset variable (F1.1)
2. Poisson model with pol_duration as the offset. (F1.2)
3. Poisson model with pol_sit_duration as the offset. (F1.3)

Tables 13, 14 and 15 in Appendix A display the results of the models. The following are the takeaways from the results:

1. From Tables 13, 14 and 15 we can see that there are some differences in estimates from the 3 models.For example, comparing frequency of payment with Biannual payment as the reference, the yearly payment has a lower number of claims for the poisson model with no offset. For the poisson models with pol_duration and pol_sit_duration offset, the yearly claim amount actually have a positive value, signifying an increase in claim number.

2. The results obtained from the three models are reasonably intuitive with respect to the fact that they fall within the ranges of expectation. For example the value for vh_age is negative across all three models, signifying that for older cars, the number of claims is lower than for newer cars.

In order to pick the best Poisson model, the AIC, loglikelihood and MSE were calculated for each model and they were compared. From Table 7, the Poisson model with no offset emerged as the best Poisson model because it had the lowest AIC, the largest loglikelihood and the smallest MSE.

Table 7: Goodness of Fit Statistics for Poisson Models

NAME_POISSON	AIC_POISSON	LOGLIK_POISSON	MSE_POISSON
No offset	58582.41	-29272.21	5.346680
Pol_Duration offset	65214.02	-32589.01	7.629141
Pol_Sit_Duration offset	65599.41	-32777.70	7.760993

3.2.2 Model F2

Model F2 is very similar to Model F1, but instead of using a Poisson distribution to model claim counts, a negative binomial distribution is used. Just like the poisson, 3 negative binomial models were fit: one with no offset, the second with **pol_duration** as offset and the third with **pol_sit_duration** as offset. The parameter estimates for these three models are in Tables 16, 17 and 18 in Appendix A.

Table 8 shows some goodness of fit statistics in order to help choose the best negative binomial model. The choice here is not so clear cut as the negative binomial model with no offset has the lowest AIC, the negative binomial model with **pol_duration** has the highest loglikelihood and the negative binomial model with **pol_sit_duration** has the lowest MSE. Since the goodness of fit statistics among the three models do not differ by much, we chose the negative binomial model with **pol_duration** as our candidate negative binomial model.

Table 8: Goodness of Fit Statistics for Negative Binomial Models

NAME_NEGBIN	AIC_NEGBIN	LOGLIK_NEGBIN	MSE_NEGBIN
No offset	58412.61	-29189.31	5.341222
Pol_Duration offset	58414.37	-29189.18	5.341018
Pol_Sit_Duration offset	58427.47	-29197.74	5.333779

3.2.3 Overdispersion Test

Overdispersion describes the observation that the variance is higher than would be expected. Some distributions like the Poisson distribution do not have a parameter to fit variability of the observation. For the poisson distribution the variance increases with the mean (i.e. the variance and the mean have the same value). So we expect the expected value and variance to be the same value. But problems arise when they are not equal. The observed variance is higher in most cases and this is termed as **overdispersion**.

Even though we have chosen our best poisson and negative binomial models already and we could directly compare them using goodness of fit statistics and 10 fold cross-validation, we would compare all initial poisson models against all initial negative binomial models as a foundation for making an informed decision on which best model (Best Poisson vs best negative binomial) is better.

A formal test for overdispersion is given as follows:

$$H_0 = E[Y] = Var[Y] = \mu$$

$$H_a = Var[Y] = \mu + \theta \times \text{trafo}(\mu)$$

Overdispersion corresponds to $\theta > 0$ and common specifications of the **trafo** function are $\text{trafo}(\mu) = \mu$ which is a negative binomial distribution with a linear variance and $\text{trafo}(\mu) = \mu^2$ which is a negative binomial distribution with a quadratic variance. The results of both specifications are displayed in Table 9. For all cases, $\theta > 0$, meaning

overdispersion exists, which makes us lean heavily towards using the negative binomial distribution in modelling claims.

Table 9: Overdispersion Test

Model	trafo	z.value	p.value	theta
No offset	1	10.704724	0	0.0781095
Pol_Duration offset	1	11.231389	0	0.5304299
Pol_Sit_Duration offset	1	14.374254	0	0.3340640
No offset	2	11.231389	0	0.5304299
Pol_Duration offset	2	5.840384	0	0.7140612
Pol_Sit_Duration offset	2	6.739414	0	0.7518428

3.2.4 Likelihood Ratio Test

The Likelihood Ratio(LR) Test is commonly used to evaluate the difference between nested models. In our case, the idea is that the poisson is a special case of the negative binomial with $\theta = 0$. A formal test is given below:

$$H_0 = \theta = 0$$

$$H_a = \theta > 0$$

This is a one-sided test, hence θ is non-negative. A p-value less than $\alpha = 0.05$ means we reject the null hypothesis and conclude that there is not enough information to say that $\theta = 0$. From table, we realize that all p-values are less than the significance level of $\alpha = 0.05$ meaning that the negative binomial model is preferred in this case as well.

Table 10: Overdispersion Test

Model	LR	p.value
No offset	165.798	0
Pol_Duration offset	6799.650	0
Pol_Sit_Duration offset	7159.934	0

3.2.4 Best Frequency Model Selection

We assesed the predictive accuracy of the two models to pick the best using the Root Mean Square Error (RMSE) statistics with 10-fold cross-validation. From the results in Table 11, there is not much to separate the models, with the best poisson model having a slight edge.

Table 11: 10-Fold CV for Frequency Models

	Best Poisson Model	Best Negative Binomial Model
RMSE	0.3939	0.394

Given that overdispersion tests favoured the negative binomial model we took a look at Rootograms for the models to further assess them. A rootogram is a diagram proposed by Kleiber and Zeileis (2016) as an improved approach to the assessment of the fit of a count regression model. Figure 2 shows the rootograms for the Poisson model and the negative binomial models side-by-side. Before delving into comparisons, we look at the main features of a rootogram:

1. Expected counts given by the model are shown by the thick red line.
2. Observed counts are shown as bars, which hang from the red line of expected counts.
3. On the x-axis, we have bins of counts. (i.e. 0 count, 1 count, etc).
4. On the y-axis we have the square root of expected count. The square root transformation exists to allow for departures from expectations to be noticed at small frequencies.
5. A reference line drawn at a height of 0.

So, the rule of thumb is that if a bar does not reach the zero line, then a model overpredicts a particular count bin and if a bar exceeds the reference line, the model underpredicts.

From Figure 2, we realize that the Poisson GLM sees a general agreement between expected and observed counts but there is an overprediction at claim count 1 as well as under-predictions at claim counts 2 and 3. Comparatively, the rootogram for the negative binomial shows a perfect agreement between expected and observed claim counts with no occurrence of over-prediction or under-prediction. Hence, we can pick the negative binomial model with **pol_duration** offset as the best frequency model.

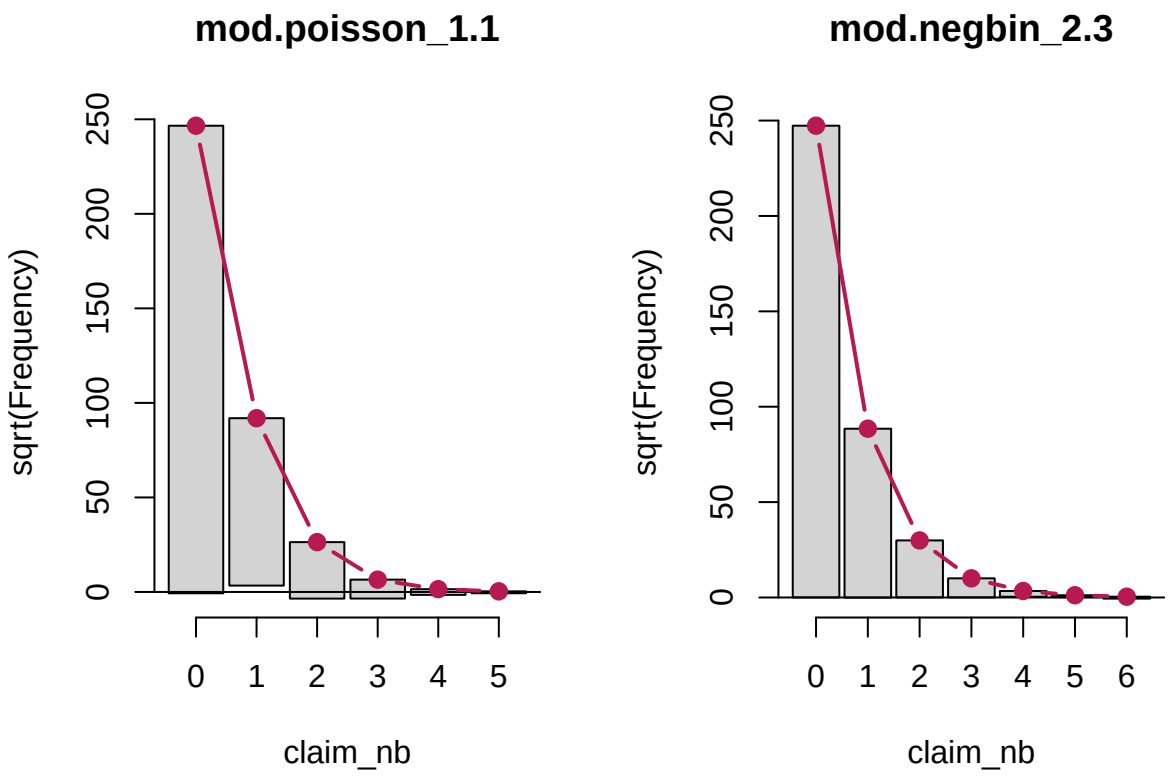

Figure 2: Rootograms For best Poisson and best Negative Binomial

3.2.5 Models S1 & S2

In Model S1, the target variable is claim amount. The assumed distribution is gamma, and there is no offset term. The model output is shown in Table 19 in Appendix A.

The primary takeaways from the model output are as follows:

1. Many of the variables that were significant in the frequency models are insignificant in this model.
2. Comparing this model solely to the best frequency model (the Negative binomial model with pol_duration as offset), we realize that **pol_bonus**, **drv_age1**, **drv_age2**, and **vh_age**, are significant to both models. So we can conclude that the ages of primary and secondary drivers are both driven by frequency and severity. Policy bonus and age of vehicle are likewise driven by frequency and severity.
3. The coefficient for **drv_age1** is positive while that for **drv_age2** is negative. So, while a unit increase in **drv_age1** increases claim amounts slightly, a unit increase in **drv_age2** decreases claim amounts.
4. Despite the reduction of the model, **vh_sale_begin** remains insignificant. This implies that **vh_sale_begin** is not an important factor in determining modelling claim amounts.

For model S2, the output is displayed in Table 20 of Appendix A. The estimates of S2 are somewhat similar to the estimates of S2 with **drv_age2**, **drv_age_lic2**, and **vh_age** being significant to claim amounts in both models.

3.2.6 Best Severity Model Selection

Table 12 makes a direct comparison of the two severity models based on the RMSE using 10-fold cross-validation. The Gamma model has a smaller RMSE. Hence we settle on the Gamma model (S1) as our severity model.

Table 12: 10-Fold CV for Severity Models

	Gamma Model	Lognormal Model
RMSE	1954.885	2323.112

3.3 Fitting Pure Premium Model

For the pure premium model, the target variable is pure premium (dollars of loss divided by policy years). The assumed distribution is Tweedie, and there is no offset. Additionally, for a Tweedie distribution, we must select the Tweedie power parameter (p), which is a function of the coefficient of variation of the underlying severity distribution. We chose 3 values of $p : 0, 1, 2$ and compared them to each other.

Additionally, we also fit a Tweedie model with a double GLM. The idea is that the Tweedie distribution can still be used to model data for which frequency and severity move in opposite directions, as long as one models dispersion as well as the mean. The assumption of positive correlation between frequency and severity stems from the assumption of a constant dispersion parameter across the dataset.However, if the dispersion (as well as the mean) varies by record, then the constraint no longer applies and assumptions are relaxed.

The outputs of the 4 models are shown in Tables 21, 22, 23 and 24 in Appendix A. The coefficients for the pure premium models are somewhat similar to the sum of the coefficients for Models F1 and S1.For example, consider the variable **drv_age1**. The sum of coefficients for Models F1 and S1 are $0.0056780 + 0.0056547 = 0.0113327$ which is quite close to the 0.0134323 for the Tweedie with $p = 2$. Generally, combining claim frequency and claim severity models should yield results that are similar to the results obtained from pure premium models.

To pick the best Tweedie model, we compared the candidate models based on the MSE. The results are displayed in Table 13. The Tweedie model with $p = 2$ has the lowest MSE among the three models. To make results comparable across the board, we used 10-fold cross-validation to calculate the RMSE for the purpose of answering our final research question.

Table 13: Goodness of Fit Statistics for Tweedie Models

Model	MSE
Tweedie (p=0)	982519.6
Tweedie (p=1)	982274.7
Tweedie (p=2)	982274.2
DGLM Tweedie	982407.2

3.4 Model Comparison: Frequency-Severity vs Pure Premium

Finally, we need to provide an answer to the question: Which modelling approach performs the better on our dataset? In order to determine this, we combined Models S1 and F2 and called it the frequency- severity model. The combined model estimates are just the addition of parameter estimates for F2 and S1. The RMSE of the combined Frequency-Severity is just a sum of the RMSEs of F2 and S1 respectively. Table 14 shows the RMSE for the combined Frequency Severity Model and the Pure premium

Table 14: Frequency-Severity vs Pure Premium

	F2-S1	Tweedie (with $p = 2$)
RMSE	1955.249	1950.948

The pure premium model has the better RMSE so it is slightly preferred over the frequency-severity model.

4. Discussion

4.1 Conclusion

The aim of this project was to develop a model for predicting pure premium/loss cost. Two approaches were analysed and compared against each other. The first being the Frequency-Severity model which involved modelling frequency and severity separately and the second being the pure premium model which involved modelling the average cost which incorporates the frequency and severity. Generalized Linear Modelling was used in building the models and the Root Mean Square Error (RMSE) using 10-fold cross validation was used as a performance metric in choosing the best model. The Pure Premium model (Tweedie model with $p = 2$) was appropriately chosen, although its performance was ever so slightly better than the Frequency-Severity model.

In practice, actuaries have primarily used the Frequency-Severity approach but currently, the pure premium model is gaining popularity. This study conducted validates that claim by picking the pure premium model over the Frequency-Severity model. Property & Casualty actuaries are therefore encouraged to build and use the Tweedie model in predictive modelling of pure premium.

4.2 Challenges

This study was not without challenges. Firstly, there was a high proportion of zero claims in the distribution of the claims number which meant that zero-inflated models were suitable to the data. However, the zero-inflated poisson and negative binomial models built in this study failed to converge. Subsequently, we fit the standard negative binomial (which accounts for overdispersion) and poisson models.

Secondly, while evaluating the prediction accuracies of the 4 Tweedie Models, we evaluated and chose the best Tweedie model by comparing their mean squared errors. Next, the best severity model was chosen by comparing the best Tweedie model against the best frequency-severity model using their root mean square errors. We did not

evaluate all 4 models on their root mean squared errors because the caret package has no in-built command for evaluating cross-validation error for double GLM Tweedie model.

4.3 Suggestions for Further Research

There were some suggestions made at the conclusion of the study. THey are as follows:

1. This study was limited to the auto-insurance line of business. To validate the industry-wide use of the Tweedie model, other lines of business in the property and casualty field can be considered in further research.
2. This data is fictitious and may not be representative of actual industry data. Thus, further studies can use actual industry data to validate results obtained from the study.
3. Further studies can find the 10-fold cross-validated RMSE for the double GLM Tweedie model by pseudocodes instead of relying on the caret package.

References

1. Zeileis A., Kleiber C., & Jackman S. (2008). **Regression Models for Count Data in R**. (n.p.).
2. Frees E. W. (2010) **Regression Modeling With Actuarial and Financial Applications**. Cambridge. Cambridge University Press.
3. Frees W., Derrig R., Meyers G. (2014). **Predictive Modeling Applications in Actuarial Science**. Vol 1. Cambridge. Cambridge University Press.
4. Frees W., Derrig R., Meyers G. (2016). **Predictive Modeling Applications in Actuarial Science**. Vol 2. Cambridge. Cambridge University Press.
5. de Jong P., Heller G. (2008). **Generalized Linear Models for Insurance Data**. Vol 2. Cambridge. Cambridge University Press.
6. James G., Witten D., Hastie T., & Tibshirani R.(2013). **An Introduction to Statistical Learning**. New York. Springer.
7. **Predictive Analytics Reshaping Insurance Industry**. Retrieved from: https://www.duckcreek.com/blog/predictive-analyitics-reshaping-insurance-industry/ on 02/21/2021.
8. **The Expanding Use of Predictive Analytics in Claims Management**. Retrieved from https://www.genre.com/knowledge/publications/iinapc1804-en.html on 01/15/2021.
9. **Predictive Modelling for Motor Insurance Claims Using Artificial Neural Networks**. Retrieved from: ID38_Pg160-172_Predictive-Modelling-for-Motor-Insurance-Claims-Using-Artificial-Neural-Networks_2.pdf (ijasca.com) on 03/12/2021.
10. No Title. Retrieved From: insurance-econometrics/[Report]econometrics-insurance.pdf at master · anhdanggit/insurance-econometrics · GitHub on 03/01/2021.

11. **Create Awesome LaTex Table with knitr::kable and KableExtra**. Retrieved from :https://haozhu233.github.io/kableExtra/awesome_table_in_pdf.pdf on 02/13/2021.

12. **6 Available Models. The caret Package**. Retrieved from: topepo.github.io/caret/available-models.html on 01/29/2021.

13. **Regression Model Validation. Cross-Validation Essentials in R**. Retrieved from: www.sthda.com/english/articles/38-regression-model-validation on 03/29/2021.

14. No title. Retrieved from: https://www.geeksforgeeks.org/repeated-k-fold... on 03/22/2021.

15. **Latex Formulas or symbols in table cells using knitr and kableExtra in R-Markdown**. Retreived From https://stackoverflow.com/questions/49416492/latex-formulas-or-symbols-in-table-cells-using-knitr on 03/23/2021.

16. **R Markdown: The Defintive Guide**. Retrieved from: https://bookdown.org/yihui/rmarkdown/beamer-presentation.html 03/26/2021.

17. No title. Retrieved from: https://rmarkdown.rstudio.com/authoring_bibliographies_and_citations.html on 03/22/2021.

18. **Bibliography management with bibtex** . Retrieved from: https://www.overleaf.com/learn/latex/Bibliography_management_with_bibtex on 03/08/2021.

19. **Using biblatex with R Markdown**. Retrieved from: https://tex.stackexchange.com/questions/449191/using-biblatex-with-r-markdown on 03/10/2021.

20. **Claims Predictive Modelling**. Retrieved from: actuariesindia.org on 03/15/2021.

5.1 Appendix A:

Poisson Model with no offset

Table 15: Summary Statistics for Poisson Model with no offset

term	estimate	std.error	statistic	p.value
(Intercept)	-2.4872126	0.1351419	-18.4044578	0.0000000
pol_bonus	0.7426548	0.1019930	7.2814286	0.0000000
pol_coverageMedian1	-0.2211876	0.0480945	-4.5990240	0.0000042
pol_coverageMedian2	-0.1656612	0.0334452	-4.9532096	0.0000007
pol_coverageMini	-1.0048698	0.0739984	-13.5796169	0.0000000
pol_sit_duration	-0.0228563	0.0060770	-3.7611243	0.0001692
pol_pay_freqMonthly	0.0716207	0.0263929	2.7136388	0.0066549
pol_pay_freqQuarterly	0.1662461	0.0614380	2.7059161	0.0068116
pol_pay_freqYearly	-0.0438199	0.0251210	-1.7443577	0.0810967
pol_paydYes	-0.1656665	0.0582592	-2.8436128	0.0044605
drv_drv2Yes	0.2286879	0.0556940	4.1061473	0.0000402
drv_age1	0.0058888	0.0018207	3.2343402	0.0012192
drv_age2	-0.0041740	0.0011094	-3.7623164	0.0001683
drv_age_lic1	-0.0045012	0.0020038	-2.2463370	0.0246824
vh_age	-0.0275740	0.0063587	-4.3364295	0.0000145
vh_cyl	0.0002111	0.0000392	5.3787037	0.0000001
vh_sale_begin	-0.0147557	0.0057478	-2.5671856	0.0102528
vh_speed	0.0000760	0.0006182	0.1228793	0.9022026
vh_value	0.0000054	0.0000022	2.5147012	0.0119133

Poisson Model with Pol_duration offset

Table 16: Summary Statistics for Poisson Model with pol duration offset

term	estimate	std.error	statistic	p.value
(Intercept)	-4.6440511	0.1253536	-37.047594	0.0000000
pol_bonus	1.9021271	0.0916055	20.764325	0.0000000
pol_coverageMedian1	-0.0904538	0.0480543	-1.882323	0.0597921
pol_coverageMedian2	-0.1465562	0.0336974	-4.349187	0.0000137
pol_coverageMini	-0.8588454	0.0741678	-11.579758	0.0000000
pol_sit_duration	-0.0884228	0.0061843	-14.298019	0.0000000
pol_pay_freqMonthly	0.2345362	0.0262994	8.917931	0.0000000
pol_pay_freqQuarterly	0.4443841	0.0615082	7.224798	0.0000000
pol_pay_freqYearly	0.1640946	0.0251604	6.521933	0.0000000
pol_paydYes	-0.2546745	0.0582450	-4.372473	0.0000123
drv_age_lic1	-0.0205745	0.0008856	-23.232559	0.0000000
vh_age	-0.0171466	0.0063549	-2.698175	0.0069721
vh_cyl	0.0002114	0.0000413	5.115267	0.0000003
vh_sale_begin	-0.0182631	0.0057380	-3.182838	0.0014584
vh_speed	0.0011493	0.0007104	1.617837	0.1056977
vh_typeTourism	-0.2247188	0.0471687	-4.764152	0.0000019
vh_value	0.0000109	0.0000022	5.025859	0.0000005
vh_weight	-0.0000522	0.0000376	-1.386735	0.1655227

Poisson Model with Pol_sit_duration offset

Table 17: Summary Statistics for Poisson Model with pol sit duration offset

term	estimate	std.error	statistic	p.value
(Intercept)	-5.4802922	0.2591756	-21.1450947	0.0000000
pol_bonus	2.3285693	0.0771791	30.1709809	0.0000000
pol_coverageMedian1	-0.0177944	0.0479808	-0.3708652	0.7107380
pol_coverageMedian2	-0.0928957	0.0336639	-2.7595064	0.0057889
pol_coverageMini	-0.7596319	0.0739721	-10.2691684	0.0000000
pol_sit_duration	-0.0991529	0.0061761	-16.0543835	0.0000000
pol_pay_freqMonthly	0.3114572	0.0259721	11.9919870	0.0000000
pol_pay_freqQuarterly	0.5095475	0.0613884	8.3003893	0.0000000
pol_pay_freqYearly	0.1519868	0.0252147	6.0277133	0.0000000
pol_paydYes	-0.3006421	0.0584488	-5.1436833	0.0000003
pol_usageProfessional	-0.2308542	0.2208788	-1.0451622	0.2959480
pol_usageRetired	-0.6064187	0.2196777	-2.7604920	0.0057714
pol_usageWorkPrivate	-0.3347591	0.2188169	-1.5298598	0.1260514
drv_drv2Yes	0.1756194	0.0547798	3.2059185	0.0013463
drv_age2	-0.0030830	0.0010855	-2.8400795	0.0045102
drv_sex1M	-0.0613884	0.0213274	-2.8783866	0.0039971
vh_age	-0.0194293	0.0063305	-3.0691716	0.0021465
vh_cyl	0.0003204	0.0000434	7.3823460	0.0000000
vh_din	-0.0028566	0.0009071	-3.1492181	0.0016371
vh_sale_begin	-0.0190598	0.0057454	-3.3174135	0.0009086
vh_speed	0.0020218	0.0008923	2.2658852	0.0234584
vh_value	0.0000128	0.0000026	4.8684770	0.0000011

Negative Binomial Model: No offset

Table 18: Summary Statistics for Negative Binomial Model with no offset

term	estimate	std.error	statistic	p.value
(Intercept)	-2.4577576	0.1401489	-17.5367544	0.0000000
pol_bonus	0.7755418	0.1074542	7.2174194	0.0000000
pol_coverageMedian1	-0.2222439	0.0495758	-4.4829080	0.0000074
pol_coverageMedian2	-0.1629295	0.0346065	-4.7080605	0.0000025
pol_coverageMini	-1.0110918	0.0751945	-13.4463524	0.0000000
pol_sit_duration	-0.0251177	0.0062579	-4.0137261	0.0000598
pol_paydYes	-0.1746522	0.0602312	-2.8996969	0.0037352
drv_drv2Yes	0.2276135	0.0581727	3.9127200	0.0000913
drv_age1	0.0056547	0.0018914	2.9896927	0.0027926
drv_age2	-0.0041574	0.0011570	-3.5932514	0.0003266
drv_age_lic1	-0.0051817	0.0020824	-2.4883747	0.0128328
vh_age	-0.0268065	0.0065856	-4.0704586	0.0000469
vh_cyl	0.0002190	0.0000410	5.3473135	0.0000001
vh_sale_begin	-0.0151340	0.0059548	-2.5414725	0.0110387
vh_speed	0.0000304	0.0006462	0.0470576	0.9624673
vh_value	0.0000052	0.0000023	2.3037724	0.0212354

Negative Binomial with Pol_duration offset

Table 19: Summary Statistics for Negative Binomial Model with pol sit duration offset

term	estimate	std.error	statistic	p.value
(Intercept)	-2.4482510	0.1414233	-17.3115137	0.0000000
pol_bonus	0.7687268	0.1083480	7.0949778	0.0000000
pol_coverageMedian1	-0.2231104	0.0496070	-4.4975593	0.0000069
pol_coverageMedian2	-0.1631054	0.0346066	-4.7131290	0.0000024
pol_coverageMini	-1.0121883	0.0752257	-13.4553456	0.0000000
pol_sit_duration	-0.0244166	0.0064141	-3.8067088	0.0001408
pol_paydYes	-0.1738070	0.0602517	-2.8846802	0.0039181
drv_drv2Yes	0.2288476	0.0582311	3.9299904	0.0000849
drv_age1	0.0056997	0.0018935	3.0101366	0.0026113
drv_age2	-0.0041824	0.0011582	-3.6112149	0.0003048
drv_age_lic1	-0.0051147	0.0020863	-2.4516348	0.0142209
vh_age	-0.0268769	0.0065872	-4.0801863	0.0000450
vh_cyl	0.0002188	0.0000410	5.3414564	0.0000001
vh_sale_begin	-0.0151118	0.0059548	-2.5377480	0.0111568
vh_speed	0.0000362	0.0006463	0.0560258	0.9553212
vh_value	0.0000052	0.0000023	2.2812544	0.0225334
log(pol_duration)	-0.0057656	0.0116197	-0.4961883	0.6197616

Negative Binomial Model with Pol_sit_duration offset

Table 20: Summary Statistics for Negative Binomial Model with pol sit duration offset

term	estimate	std.error	statistic	p.value
(Intercept)	-2.4611256	0.1400962	-17.5673958	0.0000000
pol_bonus	0.7759424	0.1075774	7.2128734	0.0000000
pol_coverageMedian1	-0.2041562	0.0493285	-4.1387034	0.0000349
pol_coverageMedian2	-0.1556491	0.0345661	-4.5029348	0.0000067
pol_coverageMini	-0.9955971	0.0750962	-13.2576185	0.0000000
drv_drv2Yes	0.2374422	0.0581948	4.0801252	0.0000450
drv_age1	0.0049988	0.0018854	2.6512894	0.0080185
drv_age2	-0.0042895	0.0011576	-3.7055734	0.0002109
drv_age_lic1	-0.0050722	0.0020826	-2.4354545	0.0148731
vh_age	-0.0292147	0.0065610	-4.4527625	0.0000085
vh_cyl	0.0002240	0.0000409	5.4716196	0.0000000
vh_sale_begin	-0.0151738	0.0059589	-2.5464073	0.0108838
vh_speed	-0.0000012	0.0006460	-0.0017903	0.9985716
vh_value	0.0000054	0.0000023	2.3825473	0.0171933
log(pol_sit_duration)	-0.0369865	0.0176979	-2.0898771	0.0366288

Gamma Model

Table 21: Summary Statistics for Gamma Model

term	estimate	std.error	statistic	p.value
(Intercept)	6.5604828	0.1851870	35.426259	0.0000000
pol_bonus	0.4965249	0.2322566	2.137828	0.0325618
drv_age1	0.0056780	0.0016744	3.391119	0.0006996
drv_age2	-0.0039908	0.0015662	-2.548015	0.0108529
drv_age_lic2	0.0075717	0.0021382	3.541119	0.0004008
vh_age	-0.0447141	0.0138980	-3.217304	0.0012993
vh_sale_begin	0.0180361	0.0127986	1.409224	0.1588091

Lognormal Model

Table 22: Summary Statistics for Lognormal Model

term	estimate	std.error	statistic	p.value
(Intercept)	7.1271816	0.0490321	145.357444	0.0000000
drv_age2	-0.0046362	0.0015744	-2.944662	0.0032427
drv_age_lic2	0.0086364	0.0021462	4.024041	0.0000578
vh_age	-0.0452167	0.0140210	-3.224921	0.0012653
vh_sale_begin	0.0198562	0.0129029	1.538893	0.1238712

Tweedie Model 1

Table 23: Summary Statistics for Tweedie Model 1

term	estimate	std.error	statistic	p.value
(Intercept)	-0.0669000	0.8953163	-0.0747222	0.9404377
pol_bonus	0.0731784	0.3117862	0.2347070	0.8144424
pol_coverageMedian1	-1.0028502	0.2602204	-3.8538490	0.0001172
pol_coverageMedian2	-1.1554405	0.1972226	-5.8585615	0.0000000
pol_coverageMini	-0.2756531	0.2981692	-0.9244854	0.3552625
pol_duration	-0.0068947	0.0035568	-1.9384518	0.0526044
pol_sit_duration	-0.0205983	0.0176951	-1.1640659	0.2444332
pol_pay_freqMonthly	-0.1274713	0.0722199	-1.7650446	0.0775957
pol_pay_freqQuarterly	-0.1639730	0.1828812	-0.8966098	0.3699551
pol_pay_freqYearly	-0.4092844	0.0679672	-6.0217915	0.0000000
pol_paydYes	-0.1165902	0.1829799	-0.6371750	0.5240297
pol_usageProfessional	0.0062818	0.7026906	0.0089396	0.9928676
pol_usageRetired	0.3709111	0.6936654	0.5347119	0.5928645
pol_usageWorkPrivate	0.1919865	0.6909909	0.2778424	0.7811408
drv_drv2Yes	0.3556958	0.1542849	2.3054481	0.0211678
drv_age1	-0.0072070	0.0065022	-1.1083848	0.2677301
drv_age2	-0.0143922	0.0023218	-6.1985968	0.0000000
drv_sex1M	-0.5820374	0.0627552	-9.2747284	0.0000000
drv_age_lic1	0.0332177	0.0065527	5.0693415	0.0000004
drv_age_lic2	0.0164794	0.0011141	14.7919829	0.0000000
vh_age	0.0620783	0.0199775	3.1074151	0.0018941
vh_cyl	0.0013058	0.0001468	8.8965390	0.0000000
vh_din	-0.0182168	0.0041322	-4.4084858	0.0000106
vh_fuelGasoline	-0.4132745	0.0928470	-4.4511359	0.0000087
vh_fuelHybrid	0.0482904	1.4670930	0.0329157	0.9737426
vh_sale_begin	0.0816139	0.0192436	4.2411030	0.0000225
vh_sale_end	-0.1777477	0.0169177	-10.5065843	0.0000000
vh_speed	0.0385034	0.0046518	8.2770836	0.0000000
vh_typeTourism	2.4324057	0.4007908	6.0690164	0.0000000
vh_value	-0.0000072	0.0000095	-0.7524615	0.4517965
vh_weight	-0.0027139	0.0002628	-10.3271240	0.0000000

Tweedie Model 2

Table 24: Summary Statistics for Tweedie Model 2

term	estimate	std.error	statistic	p.value
(Intercept)	5.2560072	0.5990178	8.7743757	0.0000000
pol_bonus	0.5346773	0.2238732	2.3883044	0.0169501
pol_coverageMedian1	-0.2853805	0.1183610	-2.4111013	0.0159276
pol_coverageMedian2	-0.4027175	0.0863948	-4.6613632	0.0000032
pol_coverageMini	0.0609249	0.1696579	0.3591045	0.7195267
pol_duration	-0.0034627	0.0029323	-1.1808943	0.2376810
pol_sit_duration	-0.0159982	0.0142671	-1.1213400	0.2621780
pol_pay_freqMonthly	0.0385505	0.0581056	0.6634562	0.5070582
pol_pay_freqQuarterly	0.1043343	0.1318463	0.7913325	0.4287742
pol_pay_freqYearly	-0.0515749	0.0557065	-0.9258319	0.3545622
pol_paydYes	-0.1121853	0.1366588	-0.8209159	0.4117195
pol_usageProfessional	0.1667493	0.4808760	0.3467615	0.7287799
pol_usageRetired	0.0787977	0.4792255	0.1644273	0.8693991
pol_usageWorkPrivate	0.1867468	0.4761827	0.3921746	0.6949400
drv_drv2Yes	0.1196798	0.1204017	0.9940043	0.3202518
drv_age1	0.0134323	0.0041706	3.2206956	0.0012841
drv_age2	-0.0066351	0.0022984	-2.8868557	0.0039019
drv_sex1M	-0.0719941	0.0483326	-1.4895568	0.1363816
drv_age_lic1	-0.0041525	0.0042785	-0.9705502	0.3318026
drv_age_lic2	0.0078810	0.0014979	5.2613974	0.0000001
vh_age	-0.0279511	0.0156975	-1.7806039	0.0750164
vh_cyl	0.0001170	0.0001159	1.0094851	0.3127736
vh_din	-0.0004913	0.0022966	-0.2139373	0.8306016
vh_fuelGasoline	0.0384861	0.0637784	0.6034348	0.5462371
vh_fuelHybrid	-0.2748856	0.7854416	-0.3499758	0.7263664
vh_sale_begin	0.0422113	0.0141453	2.9841199	0.0028528
vh_sale_end	-0.0262503	0.0146875	-1.7872573	0.0739350
vh_speed	0.0031924	0.0021883	1.4588166	0.1446562
vh_typeTourism	0.1779444	0.1234702	1.4411936	0.1495705
vh_value	0.0000056	0.0000061	0.9145365	0.3604635
vh_weight	-0.0001294	0.0000913	-1.4169359	0.1565419

Tweedie Model 3

Table 25: Summary Statistics for Tweedie Model 3

term	estimate	std.error	statistic	p.value
(Intercept)	5.3706450	0.5539573	9.6950520	0.0000000
pol_bonus	0.5366529	0.2215215	2.4225774	0.0154337
pol_coverageMedian1	-0.2724512	0.0991612	-2.7475589	0.0060180
pol_coverageMedian2	-0.3797859	0.0707635	-5.3669724	0.0000001
pol_coverageMini	0.1436581	0.1608248	0.8932581	0.3717467
pol_duration	-0.0022773	0.0027801	-0.8191623	0.4127190
pol_sit_duration	-0.0167771	0.0127846	-1.3122906	0.1894610
pol_pay_freqMonthly	0.0552870	0.0547466	1.0098705	0.3125889
pol_pay_freqQuarterly	0.1174047	0.1295208	0.9064545	0.3647236
pol_pay_freqYearly	-0.0343027	0.0522333	-0.6567202	0.5113804
pol_paydYes	-0.1045599	0.1230271	-0.8498937	0.3954105
pol_usageProfessional	0.1467329	0.4355486	0.3368921	0.7362074
pol_usageRetired	0.0250047	0.4340527	0.0576075	0.9540628
pol_usageWorkPrivate	0.1571532	0.4309868	0.3646357	0.7153933
drv_drv2Yes	0.1264754	0.1149470	1.1002937	0.2712384
drv_age1	0.0149229	0.0040699	3.6666738	0.0002474
drv_age2	-0.0064274	0.0024956	-2.5754697	0.0100288
drv_sex1M	-0.0402971	0.0456102	-0.8835103	0.3769881
drv_age_lic1	-0.0062551	0.0041944	-1.4913102	0.1359208
drv_age_lic2	0.0071586	0.0019628	3.6471648	0.0002669
vh_age	-0.0370195	0.0148224	-2.4975347	0.0125266
vh_cyl	0.0000831	0.0001090	0.7622406	0.4459397
vh_din	-0.0003036	0.0022504	-0.1349252	0.8926745
vh_fuelGasoline	0.0641462	0.0615020	1.0429932	0.2969840
vh_fuelHybrid	-0.3796440	0.6915401	-0.5489833	0.5830327
vh_sale_begin	0.0391938	0.0133616	2.9333073	0.0033636
vh_sale_end	-0.0150011	0.0136743	-1.0970296	0.2726626
vh_speed	0.0021930	0.0021035	1.0425877	0.2971718
vh_typeTourism	0.1694094	0.1044513	1.6218977	0.1048660
vh_value	0.0000085	0.0000062	1.3685302	0.1711859
vh_weight	-0.0000992	0.0000802	-1.2357830	0.2165768

Tweedie Model 4

Table 26: Summary Statistics for Tweedie Model 4

term	estimate	std.error	statistic	p.value
(Intercept)	5.2681646	0.3513629	14.9935125	0.0000000
pol_bonus	0.7371280	0.1405062	5.2462315	0.0000002
pol_coverageMedian1	-0.2883069	0.0628958	-4.5838855	0.0000046
pol_coverageMedian2	-0.3166864	0.0448837	-7.0557034	0.0000000
pol_coverageMini	0.3212915	0.1020076	3.1496814	0.0016407
pol_duration	0.0017263	0.0017633	0.9789990	0.3276111
pol_sit_duration	-0.0002995	0.0081090	-0.0369308	0.9705411
pol_pay_freqMonthly	0.0388097	0.0347246	1.1176427	0.2637543
pol_pay_freqQuarterly	-0.0233558	0.0821522	-0.2842993	0.7761886
pol_pay_freqYearly	-0.0149776	0.0331305	-0.4520810	0.6512233
pol_paydYes	-0.0233858	0.0780334	-0.2996896	0.7644220
pol_usageProfessional	-0.1279962	0.2762589	-0.4633197	0.6431483
pol_usageRetired	-0.2285612	0.2753101	-0.8301955	0.4064538
pol_usageWorkPrivate	-0.1592862	0.2733654	-0.5826861	0.5601216
drv_drv2Yes	0.0273110	0.0729083	0.3745936	0.7079730
drv_age1	0.0121584	0.0025814	4.7099276	0.0000025
drv_age2	-0.0018728	0.0015829	-1.1831140	0.2368003
drv_sex1M	-0.0161805	0.0289296	-0.5593055	0.5759694
drv_age_lic1	-0.0047878	0.0026604	-1.7996444	0.0719557
drv_age_lic2	0.0017720	0.0012450	1.4233600	0.1546721
vh_age	-0.0190101	0.0094015	-2.0220181	0.0432088
vh_cyl	0.0000767	0.0000691	1.1098057	0.2671172
vh_din	-0.0002549	0.0014274	-0.1785797	0.8582724
vh_fuelGasoline	0.1279728	0.0390094	3.2805661	0.0010406
vh_fuelHybrid	0.1217333	0.4386287	0.2775315	0.7813795
vh_sale_begin	0.0129013	0.0084750	1.5222749	0.1279811
vh_sale_end	-0.0012023	0.0086733	-0.1386242	0.8897507
vh_speed	0.0002045	0.0013342	0.1533008	0.8781650
vh_typeTourism	0.0336435	0.0662512	0.5078179	0.6115955
vh_value	0.0000103	0.0000039	2.6246746	0.0086902
vh_weight	0.0000207	0.0000509	0.4073855	0.6837362

5.2 Appendix B

R Code

```
#rm(list = ls())

knitr::opts_chunk$set(
  echo = FALSE,     # don't show code
  comment = NA,
  error=FALSE ,
  warning = FALSE,  # don't show warnings
  message = FALSE,  # don't show messages (less serious warnings)
  cache = TRUE,    # set to TRUE to save results from last compilation
  fig.align = "center"# center figures
)



library(stats)
library(MASS)
library(graphics)
library(kableExtra)
library(knitr)
library(broom)
library(tidyverse)
library(corrplot)
library(RColorBrewer)
library(lmtest)
library(PerformanceAnalytics)
library(broom)
library(lindia)
```

```
library(gridExtra)
library(ggpubr)
library(faraway)
library(caret)
library(bookdown)
library(dplyr)
library(psych)
library(ggplot2)
library(dvmisc)
library(rsq)
library(reldist)
library(cplm)
library(statmod)
library(tweedie)
library(pscl)
library(boot)
library(AER)
library(dglm)
library(countreg)
library(cplm)


options(scipen = 99)
library(kableExtra)
df=data.frame(x1=c("id_client","id_policy","id_vehicle","id_year","","",
                   "","","","","",""),
              x2=c("drv_age_lic1","drv_age_lic2","drv_age1","drv_age2",
              "drv_drv2","drv_sex1","drv_sex2","","","","",""),
              x3=c("pol_bonus","pol_coverage","pol_duration",
                  "pol_sit_duration",
```

```
                    "pol_insee_code","pol_pay_freq","pol_payd","pol_usage","",
                    "","","",
                X4=c("vh_age","vh_cyl","vh_din","vh_fuel","vh_make","vh_model",
                "vh_sale_begin","vh_sale_end","vh_speed","vh_type","vh_value",
                "vh_weight"),
                X5=c("claim_amount","claim_nb","","","","","","","","","",""))
kable(df, col.names = c("Control", "Driver", "Policy","Vehicle","Response"), booktabs= T,
caption = "Available Variables In Dataset") %>%
kable_styling(latex_options =  "hold_position")
library(dplyr)
dat1=read.csv("C:/Users/yawal/OneDrive/Desktop/Predictive/PG_2017_CLAIMS_YEAR0.csv")
dat2=read.csv("C:/Users/yawal/OneDrive/Desktop/Predictive/PG_2017_YEAR0.csv")

#sum claim_nb and claim_amount (as each client with each vehicle may claim several times)
dat1.sum = dat1 %>% group_by(id_client, id_vehicle) %>% summarize(claim_nb=sum(claim_nb),
claim_amount = sum(claim_amount))

### merge data
merged.data = left_join(dat2, dat1.sum, by=c('id_client', 'id_vehicle'))

#replace missing value by 0
i.na = is.na(merged.data$claim_nb)
merged.data$claim_nb[i.na] = 0
i.na = is.na(merged.data$claim_amount)
merged.data$claim_amount[i.na] = 0
write.csv(merged.data,"train_data.csv")
working_data=read.csv("train_data.csv")

par(mfrow=c(2,2))
```

```
hist(merged.data$claim_nb, main = paste("Histogram of claim_nb"),
     xlab = paste("claim_nb"))
hist(log(merged.data$claim_nb), main = paste("Histogram of log(claim_nb)"),
     xlab = paste("log(claim_nb)"), col="burlywood")
hist(merged.data$claim_amount, main = paste("Histogram of claim_amount"),
     xlab = paste("claim_amount"))
hist(log(merged.data$claim_amount), main = paste("Histogram of log(claim_amount)"),
     xlab = paste("log(claim_amount)"), col="burlywood")
obs=data.frame(x1=c("Number of Claims","Observation"),
               x2=c(0,87346),
               x3=c(1,11238),
               x4=c(2,1264),
               x5=c(3,134),
               x6=c(4,16),
               x7=c(5,1),
               x8=c(6,1))

kable(obs,booktabs= T,col.names = c("","","","","","","","",""),
      caption = "Number of Claims by Claim Counts") %>%
kable_styling(latex_options = c("striped","hold_position"))
des=describe(merged.data)
des.rowname=row.names(des)
des=data.frame(Mean=des$mean, SD = des$sd, Median = des$median,
               Min = des$min, Max = des$max, row.names = des.rowname)
kable(des,"latex",caption = "Summary Statistics for Variables",booktabs = T) %>%
kable_styling(latex_options = c("striped", "hold_position"))
#train and test data into 70-30
set.seed(10)
```

```
train=sample(1:nrow(working_data),0.7*nrow(working_data))
test=(-train)
train_data=working_data[train,]
test_data=working_data[test,]
attach(train_data)
#full poisson model
mod.poisson_1.0=glm(claim_nb ~pol_bonus+pol_coverage+pol_duration+
                      pol_sit_duration+pol_pay_freq ++pol_payd+pol_usage+drv_drv2+
                      drv_age1+drv_age2+drv_sex1+drv_age_lic1 + drv_age_lic2+vh_age+vh_cy
                      vh_din+vh_fuel+vh_sale_begin+vh_sale_end +
                      vh_speed+vh_type+vh_value+vh_weight, family = poisson(link = "log")
                    data = train_data, weight = NULL)

#goodness of fit statistics
mse.poisson_1.0=mean((predict(mod.poisson_1.0,test_data)-test_data$claim_nb)^2)
log.poisson_1.0=logLik(mod.poisson_1.0)
aic.poisson_1.0=mod.poisson_1.0$aic

#reduced poisson model
mod.poisson_1.1=update(mod.poisson_1.0, .~.-pol_duration-pol_usage-drv_sex1-
                         drv_age_lic2-vh_din-vh_fuel-
                         vh_sale_end-vh_type-vh_weight)

#goodness of fit statistics
mse.poisson_1.1=mean((predict(mod.poisson_1.1,test_data)-test_data$claim_nb)^2)
log.poisson_1.1=logLik(mod.poisson_1.1)
aic.poisson_1.1=mod.poisson_1.1$aic

#full poisson model with pol_duration as offset
```

```
mod.poisson_1.2=glm(claim_nb ~ pol_bonus+pol_coverage+pol_sit_duration+pol_pay_freq+pol_pa


mse.poisson_1.2=mean((predict(mod.poisson_1.2,test_data)-test_data$claim_nb)^2)

log.poisson_1.2=logLik(mod.poisson_1.2)

aic.poisson_1.2=mod.poisson_1.2$aic


#reduced poisson model with pol_duration as offset

mod.poisson_1.3=update(mod.poisson_1.2, .~.-pol_usage-drv_drv2-drv_age2-drv_sex1-drv_age1


#goodness of fit statistics

mse.poisson_1.3=mean((predict(mod.poisson_1.3,test_data)-test_data$claim_nb)^2)

log.poisson_1.3=logLik(mod.poisson_1.3)

aic.poisson_1.3=mod.poisson_1.3$aic


#full poisson model with pol_sit_duration as offset

mod.poisson_1.4=glm(claim_nb ~ pol_bonus+pol_coverage+pol_duration+pol_pay_freq+pol_payd+


#goodness of fit statistics

mse.poisson_1.4=mean((predict(mod.poisson_1.4,test_data)-test_data$claim_nb)^2)

log.poisson_1.4=logLik(mod.poisson_1.4)

aic.poisson_1.4=mod.poisson_1.4$aic


#reduced poisson model with pol_sit_duration as offset

mod.poisson_1.5=update(mod.poisson_1.2, .~.-drv_age1-drv_age1-drv_age_lic1-drv_age_lic2-vi


#goodness of fit statistics

mse.poisson_1.5=mean((predict(mod.poisson_1.5,test_data)-test_data$claim_nb)^2)

log.poisson_1.5=logLik(mod.poisson_1.5)

aic.poisson_1.5=mod.poisson_1.5$aic
```

```
#goodness of fit statistics summary
NAME_POISSON=c("No offset","Pol_Duration offset","Pol_Sit_Duration offset")
AIC_POISSON=c(aic.poisson_1.1, aic.poisson_1.3, aic.poisson_1.5)
LOGLIK_POISSON=c(log.poisson_1.1,log.poisson_1.3,log.poisson_1.5)
MSE_POISSON=c(mse.poisson_1.1,mse.poisson_1.3,mse.poisson_1.5)

res=data.frame(NAME_POISSON,AIC_POISSON,LOGLIK_POISSON,MSE_POISSON)
kable(res,"latex",
      caption = "Goodness of Fit Statistics for Poisson Models", booktabs = T) %>%
kable_styling(latex_options = c("striped", "hold_position"))
#negative binomial with no offset
mod.negbin_2.0=glm.nb(claim_nb ~ pol_bonus+pol_coverage+pol_duration+pol_sit_duration+pol

#goodness of fit statistics
mse.negbin_2.0=mean((predict(mod.negbin_2.0,test_data)-test_data$claim_nb)^2)
log.negbin_2.0=logLik(mod.negbin_2.0)
aic.negbin_2.0=mod.negbin_2.0$aic

#reduced model with no offset
mod.negbin_2.1=update(mod.negbin_2.0,.~.-pol_duration-pol_pay_freq-pol_usage-drv_sex1-drv

#goodness of fit statistics
mse.negbin_2.1=mean((predict(mod.negbin_2.1,test_data)-test_data$claim_nb)^2)
log.negbin_2.1=logLik(mod.negbin_2.1)
aic.negbin_2.1=mod.negbin_2.1$aic

#full NB model with pol_duration as offset
mod.negbin_2.2=glm.nb(claim_nb ~ pol_bonus+pol_coverage+pol_sit_duration+pol_pay_freq+pol
```

```
#goodness of fit statistics
mse.negbin_2.2=mean((predict(mod.negbin_2.2,test_data)-test_data$claim_nb)^2)
log.negbin_2.2=logLik(mod.negbin_2.2)
aic.negbin_2.2=mod.negbin_2.2$aic

#reduced NB model with pol_duration as offset
mod.negbin_2.3=update(mod.negbin_2.2,.~.-pol_pay_freq-pol_usage-drv_sex1-drv_age_lic2-vh_

#goodness of fit statistics
mse.negbin_2.3=mean((predict(mod.negbin_2.3,test_data)-test_data$claim_nb)^2)
log.negbin_2.3=logLik(mod.negbin_2.3)
aic.negbin_2.3=mod.negbin_2.3$aic

#full NB model with pol_sit_duration as offset
mod.negbin_2.4=glm.nb(claim_nb ~ pol_bonus+pol_coverage+pol_duration+pol_pay_freq+pol_pay

#goodness of fit statistics
mse.negbin_2.4=mean((predict(mod.negbin_2.4,test_data)-test_data$claim_nb)^2)
log.negbin_2.4=logLik(mod.negbin_2.4)
aic.negbin_2.4=mod.negbin_2.4$aic

#reduced NB model with pol_sit_duration as offset
mod.negbin_2.5=update(mod.negbin_2.4,.~.-pol_duration-pol_pay_freq-pol_usage-drv_sex1-drv_

#goodness of fit statistics
mse.negbin_2.5=mean((predict(mod.negbin_2.5,test_data)-test_data$claim_nb)^2)
log.negbin_2.5=logLik(mod.negbin_2.5)
aic.negbin_2.5=mod.negbin_2.5$aic
```

```r
#goodness of fit statistics summary
NAME_NEGBIN=c("No offset","Pol_Duration offset","Pol_Sit_Duration offset")
AIC_NEGBIN=c(aic.negbin_2.1, aic.negbin_2.3, aic.negbin_2.5)
LOGLIK_NEGBIN=c(log.negbin_2.1,log.negbin_2.3,log.negbin_2.5)
MSE_NEGBIN=c(mse.negbin_2.1,mse.negbin_2.3,mse.negbin_2.5)

res2=data.frame(NAME_NEGBIN,AIC_NEGBIN,LOGLIK_NEGBIN,MSE_NEGBIN)
kable(res2,"latex",
      caption = "Goodness of Fit Statistics for Negative Binomial Models", booktabs = T) %
kable_styling(latex_options = c("striped", "hold_position"))
### Overdispersion test: Nbin vs. Poisson Test ####
# LR test: alpha = 0 ####

###No offset
###p-value
p.1=pchisq(2*(logLik(mod.negbin_2.1)-logLik(mod.poisson_1.1)),df=1,lower.tail = FALSE)

### lr stat
lr.1=as.numeric(2*(logLik(mod.negbin_2.1) - logLik(mod.poisson_1.1)))

###Pol_duration offset
###p-value
p.3=pchisq(2*(logLik(mod.negbin_2.3)-logLik(mod.poisson_1.3)),df=1,lower.tail = FALSE)

### lr stat
lr.3=as.numeric(2*(logLik(mod.negbin_2.3) - logLik(mod.poisson_1.3)))

###Pol_sit_duration offset
```

```
###p-value
p.5=pchisq(2*(logLik(mod.negbin_2.5)-logLik(mod.poisson_1.5)),df=1,lower.tail = FALSE)


### lr stat
lr.5=as.numeric(2*(logLik(mod.negbin_2.5) - logLik(mod.poisson_1.5)))



###Overdispersion
AER.1.1.1=AER::dispersiontest(mod.poisson_1.1,trafo=1, alternative = "greater")
AER.1.3.1=AER::dispersiontest(mod.poisson_1.3,trafo=1)
AER.1.5.1=AER::dispersiontest(mod.poisson_1.5,trafo=1)
AER.1.1.2=AER::dispersiontest(mod.poisson_1.1,trafo=2)
AER.1.3.2=AER::dispersiontest(mod.poisson_1.3,trafo=2)
AER.1.5.2=AER::dispersiontest(mod.poisson_1.5,trafo=2)


#Overdispersion Summary
Model=c("No offset","Pol_Duration offset","Pol_Sit_Duration offset")
trafo=c(1,1,1,2,2,2)
z.value=c(AER.1.1.1$statistic,AER.1.1.2$statistic,AER.1.5.1$statistic,AER.1.1.2$statistic
p.value=c(AER.1.1.1$p.value,AER.1.1.2$p.value,AER.1.5.1$p.value,AER.1.1.2$p.value,AER.1.3
theta=c(AER.1.1.1$estimate,AER.1.1.2$estimate,AER.1.5.1$estimate,AER.1.1.2$estimate,AER.1


res4=data.frame(Model,trafo,z.value,p.value,theta)
kable(res4,"latex",
      caption = "Overdispersion Test", booktabs = T,align = "lcccc") %>%
kable_styling(latex_options = c("striped", "hold_position"))
###LR Test Summary
Model=c("No offset","Pol_Duration offset","Pol_Sit_Duration offset")
LR=c(lr.1,lr.3,lr.5)
```

```
p.value=c(p.1,p.3,p.5)


res5=data.frame(Model,LR,p.value)
kable(res5,"latex",
      caption = "Overdispersion Test", booktabs = T, align = "lcc") %>%
kable_styling(latex_options = c("striped", "hold_position"))
par(mfrow=c(1,2))
rootogram(mod.poisson_1.1, style="hanging")
rootogram(mod.negbin_2.3, style="hanging")
###10 fold cross validation
# Define training control
library(caret)
set.seed(10)
train.control=trainControl(method = "cv", number = 10)


# Train the Best poisson model


mod.fin.pois=train(claim_nb~pol_bonus+pol_coverage+pol_sit_duration+pol_pay_freq + pol_pa
,data=train_data,method="glm",trControl = train.control,na.action=na.exclude)



# Train the Best Nbin model


mod.fin.nbin=train(claim_nb ~ pol_bonus+pol_coverage+pol_duration+pol_sit_duration+pol_pa
#gamma model
mod.gamma_1.0=glm(claim_amount ~ pol_bonus+pol_coverage+pol_duration+pol_sit_duration+pol
                    pol_usage+drv_drv2+drv_age1+drv_age2+drv_sex1+drv_age_lic1+
                    drv_age_lic2+vh_age+vh_cyl+vh_din+vh_fuel+vh_sale_begin+vh_sale_end+v
```

```r
#goodness of fit statistics
mse.gamma_1.0=mean((predict(mod.gamma_1.0,subset(test_data, claim_amount>0))-test_data$cla
log.gamma_1.0=logLik(mod.gamma_1.0)
aic.gamma_1.0=mod.gamma_1.0$aic


#reduced gamma model
mod.gamma_1.1=update(mod.gamma_1.0, .~.-pol_coverage-pol_duration-pol_sit_duration-pol_pay


#goodness of fit statistics
mse.gamma_1.1=mean((predict(mod.gamma_1.1,subset(test_data, claim_amount>0))-test_data$cla
log.gamma_1.1=logLik(mod.gamma_1.1)
aic.gamma_1.1=mod.gamma_1.1$aic


#lognorm
mod.gaussian_1.1=glm(claim_amount ~ pol_bonus+pol_coverage+pol_duration+pol_sit_duration+p


#goodness of fit statistics
mse.gaussian_1.1=mean((predict(mod.gaussian_1.1,subset(test_data, claim_amount>0))-test_da
log.gaussian_1.1=logLik(mod.gaussian_1.1)
aic.gaussian_1.1=mod.gaussian_1.1$aic


#reduced lognorm model
mod.gaussian_1.2=update(mod.gamma_1.1, .~.-pol_bonus-pol_coverage-pol_duration-pol_sit_dur


###10 fold cross validation
# Define training control
set.seed(10)
train.control=trainControl(method = "cv", number = 10)
```

```
# Train the Best Gamma model
mod.fin.gam=train(claim_amount ~ pol_bonus+drv_age1+drv_age2+drv_age_lic2+vh_age+vh_sale_

# Train the Best Lognormal model

mod.fin.logn=train(claim_amount~ drv_drv2+drv_age2+drv_sex1+drv_age_lic1+drv_age_lic2+vh_
,method="glm",trControl = train.control,na.action=na.exclude)
library(cplm)
library(tweedie)
#tweedie model 0
mod.tweedie_1.0=cpglm(claim_amount~pol_bonus+pol_coverage+pol_duration+pol_sit_duration+p

#goodness of fit statistics
mse.tweedie_1.0=mean((predict(mod.tweedie_1.0,subset(test_data))-test_data$claim_amount)^

#tweedie model 1
mod.tweedie_1.1=glm(claim_amount ~ pol_bonus+pol_coverage+pol_duration+pol_sit_duration+p

#goodness of fit statistics
mse.tweedie_1.1=mean((predict(mod.tweedie_1.1,subset(test_data, claim_amount>0))-test_dat

#tweedie model 2
mod.tweedie_1.2=glm(claim_amount ~ pol_bonus+pol_coverage+pol_duration+pol_sit_duration+p

#goodness of fit statistics
mse.tweedie_1.2=mean((predict(mod.tweedie_1.2,subset(test_data, claim_amount>0))-test_dat

#tweedie model 3
mod.tweedie_1.3=glm(claim_amount ~ pol_bonus+pol_coverage+pol_duration+pol_sit_duration+p
```

```r
#goodness of fit statistics
mse.tweedie_1.3=mean((predict(mod.tweedie_1.3,subset(test_data, claim_amount>0))-test_data

#tweedie model 4 (with double glm)
mod.tweedie_1.4=dglm(log(claim_amount) ~ pol_bonus+pol_coverage+pol_duration+pol_sit_dura

#goodness of fit statistics
mse.tweedie_1.4=mean((predict(mod.tweedie_1.4,subset(test_data,claim_amount>0))-test_data

#goodness of fit statistics summary
Model=c("Tweedie (p=0)","Tweedie (p=1)","Tweedie (p=2)", "DGLM Tweedie")
MSE=c(mse.tweedie_1.1,mse.tweedie_1.2,mse.tweedie_1.3,mse.tweedie_1.4)

res7=data.frame(Model,MSE)
kable(res7,"latex",
      caption = "Goodness of Fit Statistics for Tweedie Models", booktabs = T, align=c("l
kable_styling(latex_options = c("striped", "hold_position"))

###10 fold cross validation
# Define training control
set.seed(10)
train.control=trainControl(method = "cv", number = 10)

# Train the Tweedie 3
mod.tweed.3=train(claim_amount ~ pol_bonus+pol_coverage+pol_duration+pol_sit_duration+pol
kable(tidy(mod.poisson_1.1), "latex",
      caption = "Summary Statistics for Poisson Model with no offset", booktabs = T) %>%
kable_styling(latex_options = c("striped", "hold_position"))
```

```
## Mid-Term Model


kable(tidy(mod.poisson_1.3), "latex",
      caption = "Summary Statistics for Poisson Model with pol duration offset", booktabs
kable_styling(latex_options = c("striped", "hold_position"))


kable(tidy(mod.poisson_1.5), "latex",
      caption = "Summary Statistics for Poisson Model with pol sit duration offset", book
kable_styling(latex_options = c("striped", "hold_position"))
kable(tidy(mod.negbin_2.1), "latex",
      caption = "Summary Statistics for Negative Binomial Model with no offset", booktabs
kable_styling(latex_options = c("striped", "hold_position"))
kable(tidy(mod.negbin_2.3), "latex",
      caption = "Summary Statistics for Negative Binomial Model with pol sit duration off
kable_styling(latex_options = c("striped", "hold_position"))
kable(tidy(mod.negbin_2.5), "latex",
      caption = "Summary Statistics for Negative Binomial Model with pol sit duration off
kable_styling(latex_options = c("striped", "hold_position"))
kable(tidy(mod.gamma_1.1), "latex",
      caption = "Summary Statistics for Gamma Model", booktabs = T) %>%
kable_styling(latex_options = c("striped", "hold_position"))
kable(tidy(mod.gaussian_1.2), "latex",
      caption = "Summary Statistics for Lognormal Model", booktabs = T) %>%
kable_styling(latex_options = c("striped", "hold_position"))
kable(tidy(mod.tweedie_1.1), "latex",
      caption = "Summary Statistics for Tweedie Model 1", booktabs = T) %>%
kable_styling(latex_options = c("striped", "hold_position"))
kable(tidy(mod.tweedie_1.2), "latex",
      caption = "Summary Statistics for Tweedie Model 2", booktabs = T) %>%
```

```
kable_styling(latex_options = c("striped", "hold_position"))

kable(tidy(mod.tweedie_1.3), "latex",

      caption = "Summary Statistics for Tweedie Model 3", booktabs = T) %>%

kable_styling(latex_options = c("striped", "hold_position"))

kable(tidy(mod.tweedie_1.4), "latex",

      caption = "Summary Statistics for Tweedie Model 4", booktabs = T) %>%

kable_styling(latex_options = c("striped", "hold_position"))

# this R markdown chunk generates a code appendix
```