Generalized Linear Models for Insurance Rating

M. Goldburd, A. Khare, D. Tevet & D. Guller (2020)

1 Overview of Technical Foundations

Generalized Linear Models (GLMs) are a means of modelling the relationship between a variable whose outcome we wish to predict and one or more explanatory variables. The predicted variable is called the ***target variable*** and is denoted *y*. In insurance ratemaking applications, y is typically one of the following:

- Claim frequency (i.e. claims per exposure)
- Claim severity (i.e. dollars of loss per claim)
- Pure premium (i.e. dollars of loss per exposure)
- Loss ratio (i.e. dollars of loss per dollar of premium)

For quantitative target variables, such as those above, the GLM will produce an estimate of the ***expected value*** of the outcome. For other applications, the target variable may be the occurrence or non-occurrence of a certain event, for example:

- Whether or not a policyholder will renew their policy.
- Whether a submitted claim contains fraud.

For such variables, a GLM can be applied to estimate the ***probability*** that the event will occur.

The explanatory variables, or ***predictors***, are denoted $x_1,\ldots,x_p$, where p is the number of predictors in the model. Potential predictors are any policy terms or policyholder characteristics that an insurer may wish to include in a rating plan, such as type of vehicle, age, or marital status.

The Components of the GLM

In a GLM, the outcome of the target variable is assumed to be driven by both a ***systematic component*** as well as a ***random component***. The systematic component refers to that portion of the variation in the outcomes that is related to the values of the predictors. For example, we may believe that driver age influences the expected claim frequency for a car policy. If driver age is included as a predictor in a frequency model, that effect is pasty of the systematic component. The random component is the portion of the outcome driven by causes *other than* the predictors in our model. This includes the "pure randomness" – that is, the part driven by circumstances unpredictable even in theory – as well as that which may be predictable with additional variables that are not in our model. For example, if driver age is *not* included in our model (for whatever reason), then its effect forms part of the random component.

Our goal in modelling with GLMs is to "explain" as much of the variability in the outcome as we can using our predictors. In other words, we aim to shift as much of the variability as possible away from the random component and into the systematic component. GLMs make assumptions about both components.

The Random Component: The Exponential Family

In a GLM, the target variable y is modelled as a random variable that follows a probability distribution. That distribution is assumed to be a member of the ***exponential family*** of distributions. This class of distributions is one that has certain properties that are useful in fitting GLMs. It includes the normal, Poisson, gamma, binomial, and Tweedie distributions. Selection and specification of the distribution is an important part of the model building process.

The randomness of the outcome of any particular risk (denoted y_i) may be formally expressed as:

$y_i \sim \text{Exponential}(\mu_i, \Phi)$

"Exponential" above does not refer to a specific distribution; it is a placeholder for any member of the exponential family. The terms inside the parentheses refer to a common trait shared by all distributions of the family: each member takes two parameters, μ and Φ, where μ is the ***mean*** of the distribution. Φ, the ***dispersion*** parameter, is related to the variance (but is not the variance!).

The parameter μ is of special interest as the mean represents the expected value of the outcome. The estimate of this parameter is said to be the "prediction" generated by the model – the mode's ultimate output. If no information about each record other than the outcome were available, the best estimate of μ would be the same for each record – the average of historical outcomes. GLMs allow us to use predictor variables to produce a better estimate, unique to each risk, based on the statistical relationships between the predictors and the target values in the historical data. The subscript i applied to μ in the above equation denotes that the μ parameter in the distribution is record-specific. Φ, on the other hand, is assumed to be the same for all records.

The Systematic Component

GLMs model the relationship between μ_i (the model prediction) and the predictors:

$g(\mu_i) = \beta_o + \beta_1 x_{i1} + {}_{\beta 2}x_{i2} + \ldots + \beta_p x_{ip}$

Above states that some specified *transformation* of μ_i (denoted $g(\mu_i)$) is equal to the ***intercept*** (β_o) plus a linear combination of the predictors and the ***coefficients*** ($\beta_1 \ldots \beta_p$). These values are estimated by the software used. The transformation of μ_i by the function g() is called the ***link function*** and is specified by the user. The right-hand side of the equation is called the ***linear predictor***; when calculated, it yields the value $g(\mu_i)$, which is the model prediction transformed by our specified link function. The value $g(\mu_i)$ is of little interest – our primary interest is in the value of μ_i itself. As such, after calculating the linear predictor, the model prediction is derived by applying the *inverse* of the function represented by g() to the result.

The link function g() serves to provide flexibility in relating the model prediction to the predictors: rather than requiring the mean of the target variable to be directly equal to the linear predictor, GLMs allow for a transformed value of the mean to be equal to it. However, the prediction must ultimately be driven by a linear combination of the predictors (hence the “linear” in “generalized linear model”).

So, a link function gives us more options in specifying a model, thereby providing a greater opportunity to construct a model that best reflects reality. When using GLMs to produce insurance rating plans, an added benefit is obtained when the link function is specified to be the natural log function (i.e. g(x) = ln(x)): a GLM with that specification (called a ***log link*** GLM) has the property of producing a multiplicative rating structure. So, when a log link is specified, the equation above becomes:

$\ln(\mu_{i}) = \beta_o + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}$

To derive μ_i, the inverse of the natural log function, the natural exponential function, is applied to both sides of the equation:

$\mu_i = \exp(\beta_o + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) = e^{\beta_0} \times e^{\beta_1 x_{i1}} \times \ldots \times e^{\beta_p x_{ip}}$

Multiplicative models are the most common type of rating structure used for pricing insurance, due to a number of advantages they have over other structures:

- They are simple to implement.
- Having additive terms in a model can result in negative premiums. With a multiplicative plan you guarantee positive premium.
- A multiplicative model has more intuitive appeal. It doesn’t make sense to say that having a violation should increase your auto premium by £500, regardless of whether your base premium is £1000 or £10,000. Rather, it makes more sense to say that the surcharge for having a violation is 10%.

Therefore, log link models, which produce multiplicative structures, are usually the model natural model for insurance risk.

Exponential Family Variance

There are two central moments of the exponential family distribution that are necessary to understand, as well as how they relate to the parameters.

- *Mean*. The mean of every exponential family distribution is μ.
- *Variance.* Var[y] = ΦV(μ).

The variance is equal to Φ (the dispersion parameter) times some function of μ, denoted V(μ). The function V(μ) is called the ***variance function***, and its actual definition depends on the specific distribution being used.

Distribution	**Variance Function [V(μ)]**	**Variance [ΦV(μ)]**
Normal	1	Φ
Poisson	μ	Φμ
Gamma	μ^2	$\Phi\mu^2$

Binomial	$\mu(1-\mu)$	$\Phi\mu(1-\mu)$
Negative Binomial	$\mu(1+k\mu)$	$\Phi\mu(1+k\mu)$
Tweedie	μ^p	$\Phi\mu^p$

For the normal distribution, the function $V(\mu)$ is constant, so the variance does not depend on μ. For all other distributions, $V(\mu)$ is a function of μ, and in most cases it is an increasing function. This is a desirable property in modelling insurance data, as we expect that higher-risk insureds would also have higher variance. Recall that a constraint of GLMs that we need to live with is that the Φ parameter must be a constant value for all risks. Thanks to the variance function, however, this doesn't mean the *variance* must be constant for all risks; our expectation of increasing variance with increasing risk can still be reflected in a GLM.

Remember that the variance function is not the variance. To get the actual variance, one must multiply the variance function by the estimated Φ, which in effect serves to scale the variance for all risks by some constant amount.

Variable Significance

For each predictor in the model, the GLM will return an estimate of its coefficient. Estimates are just that – estimates, and are themselves the result of a random process, since they were derived from data with random outcomes. If a different set of data were used, with all the same underlying characteristics but with different outcomes, the resulting estimated coefficients would be different.

An important question for each predictor then becomes: is the estimate of the coefficient reasonably close to the "true" coefficient? Does the predictor have *any* effect on the outcome at all? Or does the predictor have no effect (the "true" coefficient is zero) and the (non-zero) coefficient returned is merely the result of pure chance? There are several statistics to help answer these questions, among which are the *standard error*, *p-value*, and *confidence interval*.

Standard Error

The estimated coefficient is the result of a random process. The ***standard error*** is the estimated standard deviation of that random process. A small standard deviation indicates that the estimated coefficient is expected to be close to the "true" coefficient, giving us more confidence in the estimate. A large standard deviation tells us that a wide range of estimates could be achieved through randomness, making it less likely that the estimate we got is close to the true value.

Generally, large datasets will produce estimates with smaller standard errors than smaller datasets. More data allows us to "see" patterns more clearly. The standard error is also related to the estimated value of Φ: the larger the estimate of Φ, the larger the standard errors will be. This is because a larger Φ implies more variance in the randomness of the outcomes, which creates more "noise" to obscure the "signal", resulting in larger standard errors.

P-value

The ***p-value*** is derived from the standard error. For a given coefficient estimate, the p-value is an estimate of the probability of a value of that magnitude (or higher) arising by pure chance.

For example, suppose a certain variable in our model yields a coefficient of 1.5, with a p-value of 0.0012. This indicates that, if this variable's true coefficient is zero, the probability of getting a coefficient of 1.5 or higher purely by chance is 0.0012. In this case, it is therefore likely that the result reflects a real underlying effect – that the true coefficient is not zero, Such a variable is said to be ***significant***.

Tests of significance are usually framed in terms of the ***null hypothesis*** – the hypothesis that the true value of the variable in question is zero. For a p-value sufficiently small, we can reject the null hypothesis, that is, accept that the variable has a non-zero effect on the expected outcome. A common rule of thumb is to reject the null hypothesis were the p-value is 0.5 or lower. It allows for a 1-in-20 chance of a variable being accepted as significant when it is not. Since in a typical insurance modelling project we are testing many variables, this threshold may be too high to protect against the possibility of spurious effects making it into the model.

Confidence Interval

The p-value is used to guide our decision to accept or reject the null hypothesis that the true coefficient is zero; if the p-value is sufficiently small, we reject it. However, a hypothesis of zero is just one of many hypotheses that could be formulated and tested. We could hypothesize any other value and test against it, and the p-value would be inversely related to the degree to which the estimated coefficient differs from our hypothesized coefficient. It is natural to ask: what *range* of values, if hypothesized, would *not* be rejected at our chosen p-value threshold? This range is called the ***confidence interval***, and can be thought of as a reasonable range of estimates for the coefficient.

For example: for particular predictor, the GLM software returns a coefficient of 0.48, with a p-value of 0.00056 and a 95% confidence interval of [0.17, 0.79]. In this case, the low p-value indicates that the null hypothesis can be rejected. However, all null values in the range 0.17 to 0.79 are sufficiently close to 0.48 such that, if set as initial hypotheses, the data would produce p-values of 0.05 or higher. If we are comfortable with a threshold of p = 0.05 for accept/reject decisions, hypotheses of these values in that range would not be rejected, and so that range could be deemed to be a reasonable range of estimates.

Types of Predictor Variables

A ***continuous variable*** is a numeric variable that represents a measurement on a continuous scale. Examples include age, amount of insurance (in £), and population density. A ***categorical variable*** is a variable that takes on one of two or more possible values, thereby assigning each risk to a "category". The distinct values that a categorical value may take on are called ***levels***.

Treatment of Continuous Variables

Each continuous variable is input into the GLM as-is, and the GLM outputs a single coefficient for it. This results in the linear predictor holding a direct linear relationship with the value of the predictor: for each unit increase in the predictor, the linear predictor rises or declines by the value of the coefficient. If a log link was used, this results in the predicted value increasing/decreasing by some constant percentage for each unit increase in the predictor.

Treatment of Categorical Variables

The treatment here is more involved. One of the levels is designated as the ***base level***. The GLM software replaces the column in the input dataset containing the categorical variable with a series of indicator columns, one for each level of that variable *other than* the base level. Each of those columns takes on the values of 0 or 1, with 1 indicating membership of that level. Those columns are treated as separate predictors, and each receives its own coefficient in the output. The resulting dataset is called the ***design matrix***.

Weights

The dataset going into a GLM will frequently include rows that represent the averages of the outcomes of groups of similar risks rather than the outcomes of individual risks. For example, in a claim severity dataset, one row might represent the average loss amount for several claims, all with the same values for all predictor variables. Or perhaps a row in a pure premium dataset might represent the average pure premium for several exposures with the same characteristics. In such instances, it is intuitive that rows that represent a greater number of risks should carry more weight in the estimation of the model coefficients, as their outcome values are based on more data. GLMs allow the user to include a ***weight*** variable, which specifies the weight given to each record in the estimation process.

The weight variable, denoted ω_i, works as a modification to the assumed variance. Recall that the exponential family variance is of the form $Var[y_i] = \Phi V(\mu)$. When a weight variable is specified, the assumed variance for record i becomes:

$$Var\left[y_i\right]=\frac{\Phi V\left(\mu_i\right)}{\omega_i}$$

That is, the “regular” exponential family variance divided by the weight. The variance therefore holds an inverse relation to the weight.

Offsets

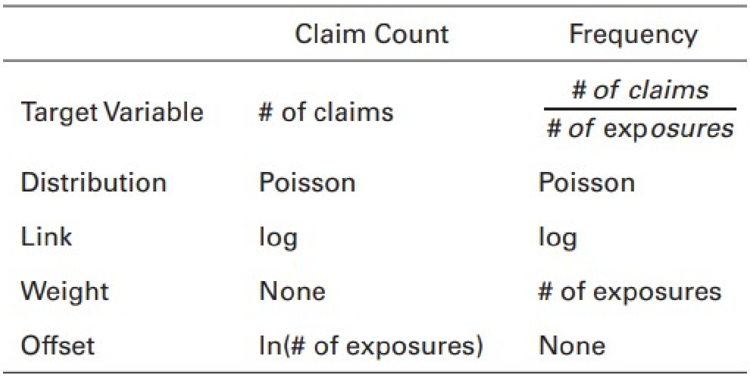
When modelling for insurance rating plans, it is often the case that the scope of the project is not to update the entire plan at once; rather, some elements will be changed while others remain as-is. In such instances, the “fixed” variable would not be assigned an estimated coefficient by the GLM. However, since it will be part of the rating plan the GLM is intended to produce, the GLM must be made aware of its existence so that the estimated coefficients for the other variables are optimal in its presence. GLMs allow you to do this through the use of an ***offset***.

An offset is formally defined as a predictor whose coefficient is constrained to be 1:

$$g(\mu_i)=\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\beta_p x_{ip}+offset$$

When including an offset in a model, it is crucial that is be on the same “scale” as the linear predictor. In the case of a log link model, this requires the offset variable to be logged prior to inclusion in the model.

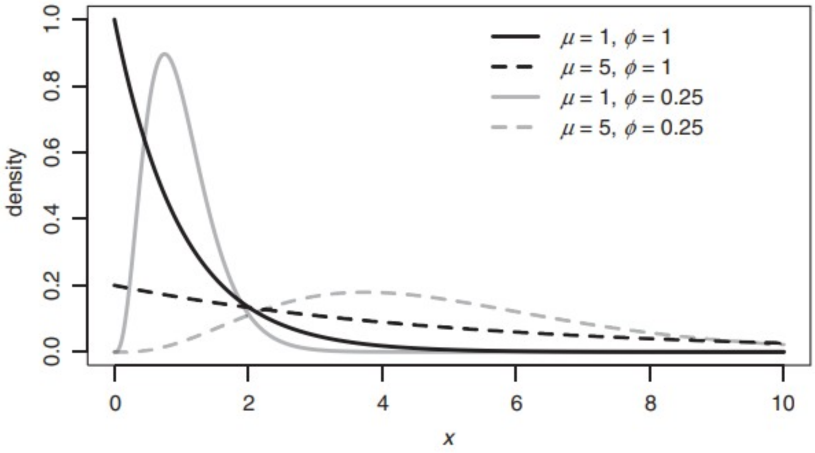
<u>Exposure Offsets</u>: Offsets are also used when modelling a target variable that is expected to vary directly with time on risk or some other measure of exposure. For example, where the target variable is the number of claims per policy where the term lengths of the policies vary; all else equal, a policy covering two car years is expected to have twice the claims as a one-year policy.



	Claim Count	Frequency
Target Variable	# of claims	$\frac{\textit{\# of claims}}{\textit{\# of exposures}}$
Distribution	Poisson	Poisson
Link	log	log
Weight	None	# of exposures
Offset	ln(# of exposures)	None

Distributions for Severity

A commonly used distribution for modelling the severity of claims is the gamma.

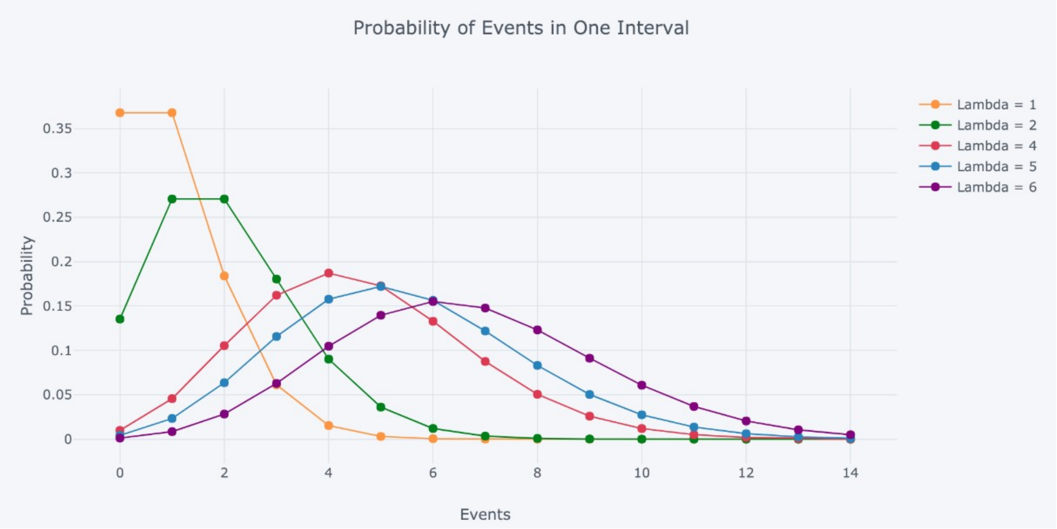


The gamma distribution is right-skewed, with a sharp peak and a long tail to the right, and it has a lower bound at zero. As these characteristics tend to be exhibited by empirical distributions of claim severity, the gamma is a natural fit for this role in a GLM. The gamma variance function is V(μ) = μ^2, meaning that the assumed variance of the severity for any claim in a gamma model is proportional to an exponential function of its mean.

The figure above shows several examples of the gamma probability density function (pdf) curves for varying values of μ and Φ. The grey lines have a lower value of Φ, so as you would expect they indicate lower variance than their corresponding black lines. The value of Φ, however, does not tell the full story of the variance. Comparing the two grey lines, gamma with μ = 5 has a much wider variance than gamma with μ = 1, despite their having the same value of Φ. This is due to the variance function, which assigns higher variance to claims with higher expected means, and is a desirable characteristic when modelling severity in a GLM.

Distributions for Frequency

The most commonly used distribution when modelling claim frequency is the ***Poisson*** distribution. The Poisson models the count of events occurring within a fixed time interval, so it is typically used for claim counts. Although the Poisson is typically a discrete distribution, its implementation in a GLM allows it to take on fractional values as well. This feature is useful for modelling claim frequency, where non-integral target variables are possible (claim count / exposure or premium).



The variance function here is V(μ) = μ, meaning that the variance increases *linearly* with the mean. In fact, in a true Poisson distribution, the variance *equals* the mean. However, claim frequency is most often found to have a variance that is greater than the mean, a phenomenon called ***overdispersion***.

A Distribution for Pure Premium: the Tweedie Distribution

Modelling pure premium (or loss ratio) at the policy level has traditionally been challenging. Consider the properties these measures exhibit, which would need to be approximated by the probability distribution used to describe them: they are most often zero, as most policies incur no loss; where they do incur a loss, the distribution of losses tends to be highly skewed. As such, the pdf would need to have most of its mass at zero, and the remaining mass skewed to the right. The ***Tweedie*** distribution can capture these properties.

In addition to the standard two exponential family parameters, the Tweedie introduces a third, p, called the ***power*** parameter. p can take on any real number except those in the interval 0 to 1 (non-inclusive: 0 and 1 themselves are valid values). The variance function for Tweedie is V(μ) = μ^p.

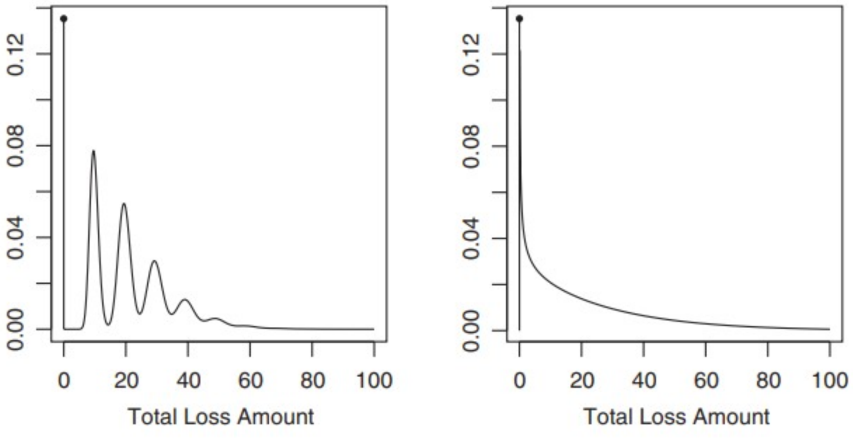
An interesting characteristic of the Tweedie is that several of the other exponential family distributions are in fact special cases of the Tweedie, dependent on the value of p:

- A Tweedie with p = 0 is a normal distribution.
- A Tweedie with p = 1 is a Poisson distribution.
- A Tweedie with p = 2 is a gamma distribution.

Thanks to the Tweedie, our choices in modelling claim severity are not restricted to the moderately-skewed gamma (or extremely skewed inverse Gaussian). The Tweedie provides a *continuum* of distributions between those two by simply setting the value of p to be between 2 (gamma) and 3 (inverse Gaussian).

The area of the p parameter space we are most interested in is between 1 and 2. At the two ends of that range are Poisson, for modelling frequency, and gamma, for modelling severity. Between 1 and 2, Tweedie becomes a neat combination of Poisson and gamma, which is great for modelling pure premium or loss ratio – that is, the combined effects of frequency and severity. [For the rest of this text, references to Tweedie refer to the specific case of a Tweedie where p is in the range [1,2]].

The Tweedie models a “compound Poisson-gamma process”. Where events (such as claims) occur following a Poisson process, and each event generates a random loss amount that follows a gamma distribution, the total loss amount for all events follows the Tweedie distribution. In this way, the Tweedie can be thought of as a “Poisson-distributed sum of gamma distributions”.



The left panel of the figure above shows an example of a Tweedie density function where p = 1.02. A value of p so close to 1 implies very little variance in the gamma (severity) component, and so the randomness of the outcome is mainly driven by the random count of events (the frequency component). As such, the shape of the distribution resembles a Poisson, with spikes at discrete points, but with a small amount of variation around each point. Note that the distribution features a point mass at 0, which allows for the likely possibility of no claims.

The right panel illustrates a Tweedie pdf for the more realistic case of p = 1.67. The gamma variation is considerably larger and therefore the discrete Poisson points are no longer visible. However, the distribution still assigns a significant probability to an outcome of 0.

The formula for the Tweedie dispersion parameter (Φ) is:

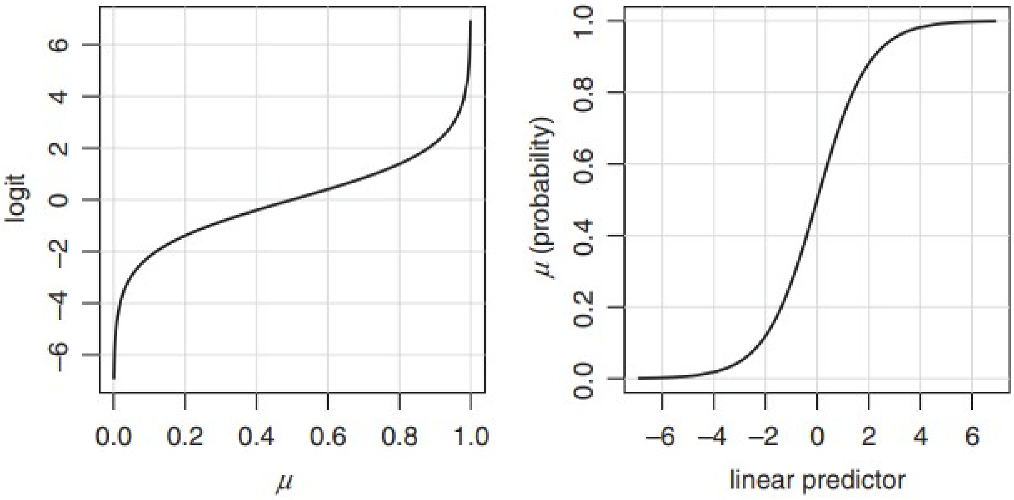
$$\Phi = \frac{\lambda^{1-p}.(\alpha\theta)^{2-p}}{2-p}$$

Through the previous 3 equations, the Tweedie parameters can be derived from any combination of the Poisson parameter (λ) and gamma parameters (α and θ) – and vice versa, with some algebraic manipulation.

Logistic Regression

For some models, the target variable we wish to predict is not a numeric value, but rather the occurrence or non-occurrence of an event. Such variables are called *binary* variables. Such a model would be built based on a datasets of historical records of similar scenarios for which the outcome is currently known. The target variable, y_i, takes on the value of 1 if the event did occur and 0 if it did not.

To model such a scenario in a GLM, the distribution of the target variables is set to be the binomial distribution. The mean of the binomial distribution – the prediction generated by the model – is the *probability* that the event will occur. A special type of link function must be used in this case. The log link cannot be used. A basic property of GLMs is that the linear predictor (right-hand side of the equation) is unbounded, and can take on any value in the range [-inf, +inf]. The mean of the binomial distribution, being a measure of probability, must be in the range [0,1]. So, we need a link function that can map a [0,1]-ranged value to be unbounded.



The most common link function for this is the ***logit*** link function, defined as:

$$g(\mu)=\ln\frac{\mu}{1-\mu}$$

The left panel in the image above shows a graph of the logit function. The logit approaches -inf as µ approaches zero and becomes arbitrarily large as µ approaches 1. The right panel shows the inverse of the logit function, the ***logistic*** function, defined as:

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

In a GLM, this function translates the value of the linear predictor onto the prediction of probability. A large negative linear predictor would indicate a low probability of occurrence, etc. A linear predictor of zero would indicate that the probability is 50%. The model can be summarized as follows:

$$\ln\frac{\mu_i}{1-\mu_i}=\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\ldots+\beta_p x_{ip}$$

The logit function can be interpreted as the log of the *odds*, where odds is defined as the ratio of the probability of occurrence to the probability of non-occurrence, or µ / (1 - µ). The odds is an alternate means of describing probability, which, unlike probability - which must lie in the region [0,1] -, is unbounded in the positive direction. (Think of a near certain event which might be said to have “million-to-one” odds).

Exponentiating both sides of the most recent equation, the logistic GLM equation becomes a multiplicative series of terms that produces the odds of occurrence. This leads to a natural interpretation of the coefficients of the GLM (after exponentiating) as describing the effect of the predictor variables on the odds. For example, a coefficient of 0.24 estimated for continuous predictor x indicates that a unit increase in x increases the odds by $e^{0.24}$ – 1, which = 27%. A coefficient of 0.24 for a categorical variable indicates that the odds for that level is 27% higher than that of the base level.

Correlation Among Predictors and Multicollinearity

Often, the predictors going into a GLM will exhibit correlation among them. Where such correlation is moderate, the GLM can handle that just fine. Determining estimates of these relativities is a strength of GLMs. It is important, before starting a modelling project, to understand the correlation structure among the predictors. This will aid in interpreting the GLM output.

Where the correlation between any two predictors is very large, however, the GLM may run into trouble. The high correlation means that much of the same information is entering the model twice. The GLM – forced not to double-count – will need to apportion the response effect between the two variables, and how to do so becomes a source of uncertainty. As such, coefficients behave erratically; one might see extremely high or low coefficient results in such scenarios. Furthermore, the standard errors associated with those coefficients will be large, and small perturbations in the data may swing the coefficient estimates wildly. Such a model is said to be *unstable*.

Such instability should be avoided. One should look out for high correlation prior to modelling, by examining two-way correlation tables. Where high correlation is detected, there are ways to deal with this:

- For any group of correlated predictors, remove all but one from the model. A potential downside is that there may be some unique information, distinct from the common information, contained in individual predictors that will not be considered in our modelling process.
- Pre-process the data using dimensionality-reduction techniques such as *PCA* or *factor analysis*. These methods create multiple new variables from correlated groups of predictors. Those new variables exhibit little or no correlation between them – making them more useful in a GLM – and they may be representative of the different components of underlying information making up the original variables.

Simple correlation between pairs of predictors are easy enough to detect using a correlation matrix. A more subtle problem may exist where two or more predictors in a model may be strongly predictive of a third, a situation known as ***multicollinearity***. The same instability problem as above may result, since the information contained in the third variable is also present in the model in the form of the *combination* of the other two variables. However, the variable may not be highly correlated with either of the other two predictors *individually* and so this effect will not show up in a correlation matrix, making it difficult to detect.

A useful statistic for detecting multicollinearity is the ***variance inflation factor*** (VIF). The VIF for any predictor is a measure of how much the (squared) standard error for the predictor is increased due to the presence of collinearity with other predictors. It is determined for each predictor by running a linear model setting the predictor as the target and using all the *other* predictors as inputs, and measuring the predictive power of that model. A rule of thumb: a VIF of greater than 10 is considered high. However, where the VIF is high, look deeper into the collinearity structure in order to make an informed decision about how best to handle it in the model.

Limitations of GLMs

1. GLMs Assign Full Credibility to the Data

The estimates produced by the GLM are fit under the assumption that the data are fully credible for every parameter.

2. GLMs Assume the Randomness of Outcomes is Uncorrelated

An assumption is that the random component of the outcome of the target variable is uncorrelated among the records in the training set. Note the qualification “random component” – this is not the same thing as saying the outcomes are uncorrelated. If our car severity model contains driver age and territory as predictors, we expect that drivers of similar ages or in the same territory would have similar outcomes, and thus be correlated in that way. After all, identifying and capturing such correlations is the point of our modelling exercise. However, the assumption is that the *random* component of the outcome, which means the portion of the outcome driven by causes not in our model – are independent.

This assumption may be violated if there exist groups of records that are likely to have similar outcomes, perhaps due to some latent variable not capture by our model:

- When modelling a line that includes a wind peril, policyholders in the same area will likely have similar outcomes, as the losses tend to be driven by storms that affect multiple insureds at the same time.

2 The Model-Building Process

The technical details of model construction are important. However, it is important to understand other steps involved in the construction and evaluation of a predictive model.

Setting Objectives and Goals

Before collecting any data or building any models, it is important to develop a clear understanding and to gain alignment on the scope and goals of the project. Important questions to ask:

- What are the goals of the analysis?
- Given the goals of the project, what is the appropriate data to collect/use? Will obtaining this data be costly and time-consuming?
- What is the time frame for completing this project?
- What are the key risks that may arise during the project and how can these be mitigated?
- Who will work on the project?

Communicating with Key Stakeholders

A common reason for a project failing or falling behind schedule is a lack of alignment on the goals/outcomes of the project among its key stakeholders. The modeler isn’t just creating a predictive model, but rather constructing a new product that will hopefully enter the market. So, key stakeholders may include:

- *Regulators*. In a modelling project, one should include all the variables that are predictive and add lift to the model. However, many variables are considered off limits in pricing insurance risk due to legal and regulatory considerations. It is important to understand these limitations.
- *IT*. The model results will likely need to be coded into a new rating system and IT systems generally have limitations. Before and during model construction, it is important to communicate the desired rating structure to programmers who will be coding the rating changes. Some components may not be feasible from an IT perspective, so one should adjust the models accordingly.
- *Underwriters*. Once the models are complete and turned into a product, someone will have to sell that product. If the new rating structure isn’t understood by the policy producers, it may be difficult to meet sales goals. By including them in the discussion, the final product can better reflect their needs and concerns, which may lead to a better business outcome.

Collecting and Processing Data

Often the most time-consuming component of a predictive modelling project. Most data is messy, so time must be spent figuring out how to clean the data, impute missing values, etc. Collecting and processing data are often an iterative process. The data should be split into at least two subsets, so that the model can be tested on data that was not used to build it.

Conducting Exploratory Data Analysis

EDA will help the modeler better understand the nature of the data and the relationships between the target and explanatory variables. Helpful plots include:

- Plotting each variable versus the target variable to see what (if any) relationship exists.
- Plotting continuous variables against each other, to see the correlation between them.

Specifying Model Form

- What type of predictive model works best for this project and this data?
- What is the target variable and which features should be included?
- Should transformations be applied to the target variable or to any of the features?
- Which link function should be used?

Evaluating Model Output

This involves assessing the overall fit of the model and identifying areas in which the model fit can be improved. One should also analyze the significance of each predictor variable, and thus removing any variables accordingly. Lastly, one can compare the lift of a newly constructed model over the existing model or rating structure.

Translating the Model into a Product

In the insurance industry, this product is often a rating plan. Considerations:

- Is the product clear and understandable? There should be no ambiguity in the risk classification. A knowledgeable person should be able to clearly understand the structure of the product.
- Are there items included in the product that were not included in the model? E.g. there are often rating factors included in the plan that are not part of the model because there is no data available on that variable. It is important to understand the potential relationship between this variable and other variables that were included in the model. It may be appropriate to apply judgmental adjustments to the variables in the rating plan.

Maintaining and Rebuilding the Model

The predictive accuracy of any model tends to decrease over time as the data used to construct the model becomes less relevant. Models should be periodically rebuilt in order to maximize their predictive accuracy, but sometimes it may be beneficial to simply refresh the existing model using newer data.

3 Data Preparation and Considerations

Although every organization has different processes for collecting, storying, and retrieving the data needed to build a rating plan, there are some common themes and situations. Remember: the data preparation step is iterative. Correcting one error often helps you discover another, and insights gleaned from the model-building process might prompt you to step back and revisit your approach to data preparation.

Modifying the Data

Check for duplicate records. Delete the duplicates.

Check numerical fields for unreasonable values. E.g. a driver age of 150 or a quote offer of £100,000.

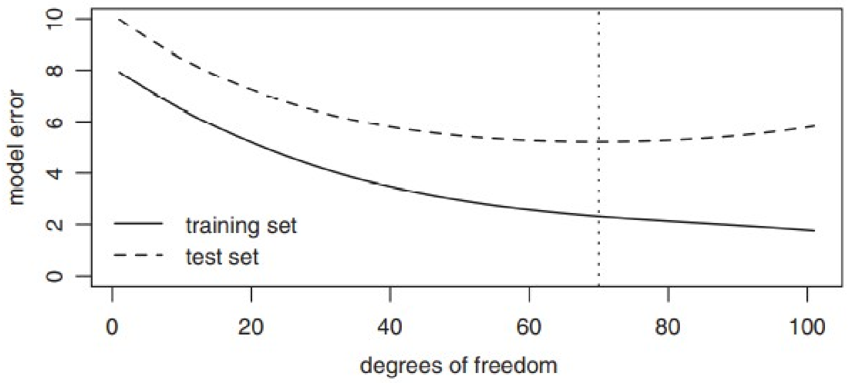
Decide how to handle each error or missing value discovered. Deleting errors is most likely the best approach – you may be left with too little data. A better solution might be to replace erroneous or missing values with the mean or modal field value, and add a new field for an error flag. The error flag can be included in the model and will proxy for the presence of an error.

Splitting the Data

The available data should be split into at least two groups. One is called the ***training set***. This is used to perform all the model-building steps – selecting the variables, determining the appropriate variable transformations, choosing the distribution, etc. Another group of data is called the ***test set***, which will be used to assess the performance of the model and may also be used to choose among several candidate models.

We do this because attempting to test the performance on any model on the same data on which the model was built will produce over-optimistic results. After all, the model-fitting process optimizes the parameters to best fit the data used and train it, so we would expect it to perform better on this data than any other. Using the training data to compare our model to any model built on different data would give our model an unfair advantage. Another reason is that as we increase the complexity of our model, the fit to the training data will *always* get better. For data the model fitting process has not seen, additional complexity may not improve the performance of a model – it may make it worse.

For a GLM, model complexity is measured in terms of ***degrees of freedom***, or the number of parameters estimated by the model-fitting procedure. Every continuous variable we include adds a degree of freedom. For a categorical variable, a degree of freedom is added for each non-base level. Furthermore, every polynomial term or interaction term – basically anything for which the model will need to estimate another parameter value – counts as a degree of freedom. Each degree of freedom provides the model more freedom to fit the training data. Since the fitting procedure always optimizes the fit, additional flexibility to fit the data better means the model *will* fit the data better.



The figure above illustrates the relationship between the degrees of freedom and the performance of the model on the training set and test set. Model performance here is measured by model error, or the degree to which the predictions “miss” the actual values, with lower error implying better model performance. The performance on the training set is always better than on the test set. Increasing the complexity of the model improves the performance on both training and test set – up to a point. Beyond that point, the performance on the training set continues to improve, but things get worse on the test set.

This is because, with enough flexibility, the model is free to “explain” the randomness in the training set outcomes (the ***noise***) in addition to the part of the outcome driven by the systematic effects (the ***signal***). The noise in the training data would obviously not generalize to new data, so this information in our model becomes a liability. A model that includes significant random noise in its parameter estimates is said to be ***overfit***.

Our goal in modelling is to find the right balance where we pick up as much of the signal as possible with minimal noise, represented by the dotted line in the picture above. So, it is critical to retain holdout data on which to test the resulting models. This out-of-sample testing allows for a truer assessment of the model’s predictive power. This division of data will remain intact throughout the entire modelling process. There are different approaches to splitting.

Train and Test

The simplest split is to create two subsets of the data, the *training set* and the *test set*. The training set should be used for the entire model building process, beginning with the initial exploration of variables all the way through the model refinement. The test set is used when the model building is complete, to compare the resulting model against the existing plan and assess the relative performance of several candidate models. Typical proportions for this split are 60/40 or 70/30 training to test. Choice of split percentages involves a trade-off. More data available for the training set will allow for clearer views of patterns in the data. However, if too little data is left for the holdout, the final assessment of models will have less certainty.

The split can be performed either by randomly allocating records between the two sets or by splitting on the basis of a time variable. This latter is usually the optimal approach when predicting ahead in time.

Validation Sets

Sometimes, you may want to throw in a validation set for hyperparameter tuning. You might train your model on the train set and assess its performance on the validation set. The model will be adjusted based on its performance on the validation data. Final performance metrics are judged based on performance on the test set. 40/30/30 – 60/20/20 could be considered for this.

The caution here is to use the test set *sparingly*. If too frequent reference is made to the test set or if too many choices of models are evaluated on it, it becomes less a test set and more of a training set; once a large part of the modelling decision has been based on how well it fits the test, that fit becomes less indicative of how the model will behave on data that it has truly not seen. For this reason, a validation set is useful (but again, don't overuse it).

Once a final model is chosen, we go back and rebuild it using all of the data, so that the parameter estimates would be at their most credible.

Cross Validation

Cross validation provides a means of assessing the performance of the model on unseen data through multiple splits of train and test. The most widely used version is ***K-fold***:

1. Split the data into *k* groups (a common choice is 5 or 10). Each group is called a *fold*. The split is done randomly or through a temporal variable.
2. For the first fold: train the mode using the other k-1 folds and test the model using the first fold.
3. Repeat step 2 for each of the remaining folds.

The output of these procedure is *k* estimates of model performance, each of which was assessed on data that its training procedure has not seen. Several models can be compared by running the procedure for each of them on the same set of folds and comparing their relative performances for each fold. This method is superior to the train/test split, since all of the data is being used to test out-of-sample model performance.

4 Selection of Model Form

Selecting the form of a predictive model is an iterative process, often more an art than a science. The model form is likely to evolve based on an analysis of the results of preliminary models.

Choosing the Target Variable

There are usually several options for the target variable. When modelling a rating plan, the target variable might be pure premium, claim frequency, or claim severity. If instead the goal of the project is to identify deficiencies in the existing rating plan, loss ratio may be a more appropriate target. The decision of which target variable to choose generally comes down to data availability and other factors.

Frequency/Severity vs Pure Premium

Where the ultimate goal of a model is to predict pure premium, there are two common approaches:

- Build two separate models: one with claims frequency (the count of claims per exposure) as the target, and another targeting claim severity (dollars of loss per claim). The individual models are then combined to form a pure premium model. Assuming log links were used for both, this combination is achieved by multiplying their corresponding relativity factors together.
- Build a single model targeting pure premium (dollars of loss per exposure) using the Tweedie distribution.

Time constraints may be a factor here, since the former approach requires building two models rather than one, especially if a large number of pure premium models must be produced (e.g. different perils). Another constraint is the data available.

The frequency/severity approach confers a number of advantages over pure premium modelling:

- Modelling frequency and severity separately often provides more insight than a pure premium model as it allows us to see the extent to which the various effects are frequency-driven versus severity-driven. This information could be useful in the model refinement process. Furthermore, some effects may get “lost” when viewed on a pure premium basis due to counteracting effects on frequency and severity. Knowledge of such a variable’s underlying effects could be useful in other business decisions.
- Each of frequency and severity is more stable – they exhibit less random variance – than pure premium. Separating out those two sources of variance from the pure premium data effectively “cuts through the noise”, enabling us to see effects in the data we otherwise would not.
- Pure premium modelling can lead to overfitting. If a variable affects frequency but not severity, if that variable does get included in our pure premium model, the model is forced to fit its coefficient to both the frequency and severity effects observed in the training data. To the extent the severity effect is spurious, the parameter is overfit.
- The distribution used to model pure premium – the Tweedie – contains the implicit assumption that frequency and severity “move in the same direction”. Where a predictor drives an increase in the target variable (pure premium or loss ratio), that increase is made up of an increase in both its frequency and severity components. Modelling frequency and severity separately frees us from this restriction.

Policies with Multiple Coverages and Perils

Even if the rating plan must be structured on an “all perils combined” basis, there is benefit to modelling the perils separately, as that will allow us to tailor the models to the unique characteristics of each peril. We can always combine the models at the end:

- Use the by-peril models to generate predictions of expected loss due to each peril for some set of exposure data.
- Add the peril predictions together to form an all-peril loss cost for each record.
- Run a model on that data, using the all-peril loss cost as the target, and the union of all the individual model predictors as the predictors.

The coefficients for the resulting model will yield the all-peril relativities implied by the underlying by-peril models for the mix of business in the data.

Transforming the Target Variable

- For pure premium, loss ratio, or severity models, the presence of a few very large losses can have undue influence on the model results. In such cases, *capping* losses at a selected large loss threshold may yield a more robust and stable model. The cap point should be high enough so that the target variable still captures the systematic variation in severity among risks, but not too high such that random large losses create excessive noise.
- Important to look out for catastrophic events that would cause a large number of losses at once, which can skew both frequency and severity effects. If possible, losses related to such events should be removed from the data entirely – limiting the scope of the model to predicting *non-catastrophic* loss only – and a catastrophe model should separately be used to estimate the effect of catastrophes on the rating variables. If not possible, the effect of these losses should be tempered by either adjusting the value of the target variable downward or by decreasing the weight, so these events do not unduly influence the parameter estimates.

Variable Selection

Choosing which variables to include in the model. A major criteria on whether to include a feature in the model is variable significance – we would like to be confident that the effect of the variable indicated by the GLM is the result of a true relationship between that predictor and the target, and not due to noise in the data. Here, we are guided by the p-value. In addition to statistical significance, other considerations for variable selection:

- Will it be cost-effective to collect the value of this variable when writing new and renewal business?
- Does inclusion of this variable in a rating plan conform to actuarial standards of practice and regulatory requirements?

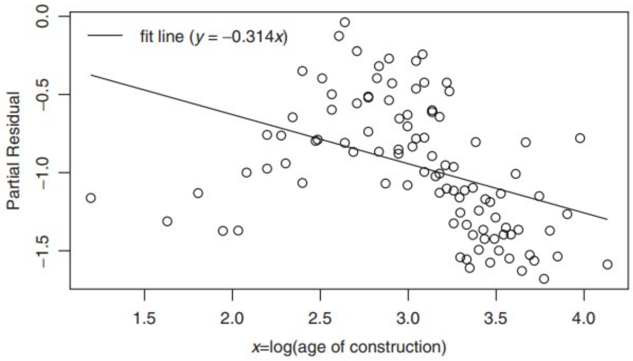
Transformation of Variables

Deciding whether or not to include a variable is often not the end of the story. The variable often needs to be transformed in some way such that the resulting model is a better fit to the data.

When including a continuous variable in a log-link model, the model assumes a linear relationship between the log of the variable and the log of the mean of the target variable. However, this relationship doesn't always hold; some variables have a more complex relationship with the target variable that cannot be described by a straight line. Here, it is necessary to transform the variable in some way so that it can adequately model the effect.

Detecting Non-Linearity with Partial Residual Plots

The ***partial residual*** may be thought of as the actual value with all components of the model prediction other than the part driven by x_j subtracted out. The variance in the partial residuals therefore contains the variance unexplained by our model in addition to the portion of the variance our model intends to explain with $\beta_j x_j$. We can plot them against the model's estimate of $\beta_j x_j$ to see how well it did.



For example, in the figure above – the model's linear estimate of the building age effect (-0.314x) is superimposed over the plot. While the line may be the best *linear* fit to the points, it is certainly not the best fit, as the points are missing the line in a systematic way. The model is clearly over-predicting for risks where log building age is 2.5 (12 in real terms) and lower. It underpredicts between 2.5 and 3.5, and once again over-predicts for older buildings. We need something more flexible than a straight line to properly fit this data.

Adding Polynomial Terms

Another means of accommodating non-linearity in a linear model is to include the square, cube, or higher-order polynomials of the variable in the model in addition to the original variable. In such a model, the original variable and the polynomial terms are all treated as separate predictors, and a separate coefficient is estimated for each. This enables the model to fit curves to the data; the more polynomial terms that are provided, the more flexible the fit that can be achieved.

One potential downside to using polynomials is the loss of interpretability. From the coefficients alone it is difficult to discern the shape of the curve; to understand the model's indicated relationship of the predictors to the target it may be necessary to graph the polynomial function. Another drawback is that polynomial functions tend to behave erratically at the edges of the data, providing unreasonable predictions at those extremes.

5 Model Refinement

We get a number of statistical measures of how well the model fits the training data, which are useful when comparing candidates for model specifications and assessing the predictive power of individual variables. The most important such measures are *log-likelihood* and *deviance*.

Log-Likelihood

For any given set of coefficients, a GLM implies a probabilistic mean for each record. That, along with the dispersion parameter and chosen distributional form, implies a full probability distribution. It is therefore possible to calculate, for any record, the probability (or probability density) that the GLM would assign to the actual outcome that has indeed occurred. Multiplying those values across *all* records produces the probability of all the historical outcomes occurring; this value is called the ***likelihood***.

A GLM is fit by finding the set of parameters for which the likelihood is the highest. This makes sense; the best model is the one that assigns the highest probability to the historical outcomes. Since likelihood is usually an extremely small number, the log of likelihood, or ***log-likelihood***, is usually used instead to make working with it more manageable. It is by itself difficult to interpret, therefore it is useful to relate it to its hypothetical upper and lower bounds achievable with the given data.

At the low end of the scale is the log-likelihood of the ***null model***, a hypothetical model with no predictors – only an intercept. Such a model will produce the same prediction for every record: the grand mean. At the other extreme likes the ***saturated model***, a hypothetical model with an equal number of predictors as there are records in the dataset. Such a model would perfectly “predict” every historical outcome. It would also be most likely useless as it would overfit to the extreme; it would be nothing more than a complicated way of restating the historical data. However, since predicting each record perfectly is the theoretical best a model can possibly do, it provides a useful upper bound to log-likelihood for this data.

While the null model yields the lowest possible log-likelihood, the saturated model yields the highest; the log-likelihood of your model will lie somewhere in between. This leads to another measure of useful model fit: deviance.

Deviance

Scaled deviance for a GLM is defined as:

$$scaled\,deviance = 2 \times \left(¿_{saturated} - ¿_{model} \right)$$

where $ll_{saturated}$ is the log-likelihood of the saturated model, and ll_{model} is the log-likelihood of the model being evaluated. More formally stated as:

$$D^{¿} = 2 \times \sum_{i=1}^{n} \ln f\left(y_i \middle| \mu_i = y_i\right) - \ln f\left(y_i \middle| \mu_i = \mu_i ¿¿\right.$$

The first term after Σ is the log of the probability of outcome y_i given that the model’s predicted mean is y_i – the mean that would be predicted by the saturated model. The second term is the log probability assigned to the outcome y_i by the actual model. The difference between those two values can be thought of as the magnitude by which the model missed the “perfect” log-likelihood for that record. Summing across all records and multiplying the result by 2 yields the scaled deviance.

Multiplying the scaled deviance by the estimated dispersion parameter Φ yields the *unscaled deviance*. This has the additional property of being independent of the dispersion parameter, making it useful for comparing models with different estimates of dispersion.

The fitted GLM coefficients are those that minimize deviance. This is equivalent to maximizing log-likelihood. The deviance for the saturated model is zero, while the deviance for the null model can be thought of as the total deviance inherent in the data. The deviance for your model will lie between those two extremes.

Limitations on the Use of Log-Likelihood and Deviance

Firstly, when comparing two models using log-likelihood or deviance, the comparison is only valid if the datasets used to fit the two models are exactly identical. Recall that the total log-likelihood is calculated by summing the log-likelihoods of the individual records across the data; if the data used for one model has a different number of records than other, the total will be different in a way that has nothing to do with model fit.

For any comparisons of models that use deviance, in addition to the caveat above, it is also necessary that the assumed distribution must be identical as well. Deviance is based on the amount by which log-likelihood deviates from the “perfect” log-likelihood; changing any assumptions other than the coefficients would alter the value of the “perfect” log-likelihood as well as the model log-likelihood, muddying the comparison.

Comparing Candidate Models

The process of building and refining a GLM is one that takes place over many iterations; frequent decisions need to be made along the way, such as which predictors to include, the appropriate transformations, etc. This next section looks are statistical tests, based on the measures of model fit just discussed, that can be used to compare successive model runs and guide our decision making.

Nested Models and the F-Test

Where a model uses a subset of the predictors of a larger model, the smaller model is said to be a ***nested model*** of the larger one. Comparisons of nested models occurs when considering to add or subtract predictors. How to find out whether a larger model or smaller model is better?

We can find out by comparing the deviance. Note: adding predictors to a model *always* reduces deviance, whether the predictor has any relation to the target variable or not. This is because more predictors – more parameters available to fit – gives the model fitting process more freedom to fit the data, and so it *will* fit the data better. Therefore, the meaningful question when comparing deviances is: did the added predictors reduce the deviance *significantly more* than we would expect them to if they are *not* predictive? A way to answer this is through the ***F-Test***, wherein the ***F-statistic*** is calculated and compared against the ***F distribution***. The formula for the F-statistic is:

$$F=\frac{Ds-Db}{\left(\text{¿}\,of\ added\ parameters\right)\times\Phi b}$$

Here, “D” refers to the unscaled deviance, “s” and “b” to “small” and “big” models respectively. The numerator is the difference in the unscaled deviance between the two models – that is, the amount by which the unscaled deviance was reduced by the inclusion of the additional parameters. This value is always positive, since the deviance always goes down. The Φb in the denominator is the estimate of the dispersion parameter for the big model. Multiplying this by the number of added parameters gives us the total expected drop in deviance. For added predictors to “carry their weight”, they must reduce deviance by significantly more than this amount.

Thus, the ratio in the equation above has an expected value of 1. If it is significantly greater than 1, we may conclude that the added variables do indeed improve the model. How much greater than 1 is significant? The F-statistic follows an F distribution, with a numerator degrees of freedom equal to the number of added parameters and a denominator degrees of freedom equal to n – pb, or the number of records minus the number of parameters in the big model.

For example, we have an auto GLM build on 972 rows of data with 6 parameters, yielding an unscaled deviance of 365.8 and an estimated dispersion parameter of 1.42. We wish to test the significance of an additional potential predictor: a categorical variable with 5 levels. We run the GLM with the inclusion of this predictor, adding 5 – 1 = 4 parameters to the model (a categorical variable with m levels adds m – 1 parameters, one for each other than the base level). Suppose the unscaled deviance is 352.1 and its estimated dispersion parameter is 1.42. The F-statistic is thus: (365.8 – 352.1) / (4 × 1.42) = 2.412.

To assess the significance of this value, we compare it against an F distribution with 4 numerator degrees of freedom and 972 – 10 = 962 degrees of freedom. An F distribution with those parameters has 2.412 at its 95.2 percentile, indicating a 4.8% probability of a drop in deviance of this magnitude arising by pure chance. As such, this predictor is found to be significant at the 95% significance level.

Penalized Measures of Fit

The F-test is only applicable to nested models. Often, we want to compare non-nested models – that is, models having different variables, where one does not contain a subset of the variables of the other. The deviance alone cannot be used, since adding parameters always reduces deviance, and so selecting on the basis of lowest deviance gives an unfair advantage to the model with more parameters, which can lead to overfitting.

A practical away to avoid the problem of overfitting is to use a *penalized measure of fit***.** While deviance is strictly a measure of model goodness of fit on the training data, a penalized measure of fit also incorporates information about the model’s complexity, and so becomes a measure of model quality. Using one of these measures, one can compare two models that have different numbers of parameters. The two most commonly used measures of deviance are ***AIC*** and ***BIC***.

AIC, or the ***Akaike Information Criterion***, is defined as:

$$AIC = -2 \times \log - likelihood + 2p$$

where p is the number of parameters in the model. As with deviance, a smaller AIC suggests a "better" model. The first term in the equation above declines as model fit on the training data improves; the second term, called the *penalty term*, serves to increase the AIC as a "penalty" for each added parameter. Using this criterion, models that produce low levels of deviance but high AICs can be discarded.

BIC, or the ***Bayesian Information Criterion***, is defined as -2 × log-likelihood + p log(n), where p is once again the number of parameters, and n is the number of data points that the model is fit on. As datasets tend to be large, the penalty for additional parameters imposed by BIC tends to be much bigger than the penalty imposed by AIC. Both AIC and BIC are good, but generally AIC is used. Relying too much on BIC may result in the exclusion of predictive variables from your model.

Residual Analysis

A useful and important means of assessing how well the specified model fits the data is by visual inspection of the *residuals*, or measures of the deviations of the individual datapoints from their predicted values. For any given record, we can think of the residual as measuring the magnitude by which the model prediction "missed" the actual value. In our GLM, this is assumed to be the manifestation of the *random* component of the model – the portion of the outcome driven by factors other than the predictors. Therefore, it is natural to inspect these values to determine how well our model actually does at capturing this randomness.

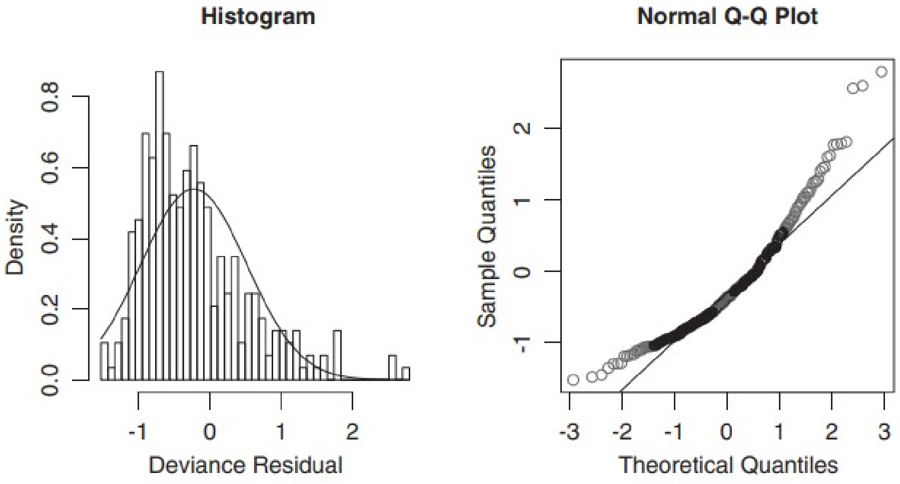
The simplest kind of residual is the ***raw residual*** – the difference between the actual and expected, or y_i - μ_i. For GLMs, two measures of deviation are more useful: the ***deviance residual*** and the ***working residual***.

Deviance Residuals

Intuitively, we can think of the deviance residual as the residual "adjusted for" the shape of the assumed GLM distribution, such that its distribution will be approximately normal if the assumed GLM distribution is correct. In a well-fit model, we expect deviance residuals to:

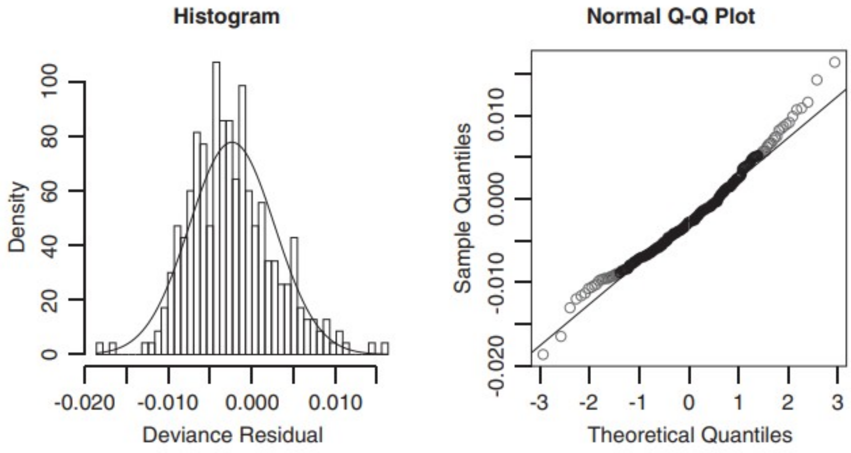
- *Follow no predictable pattern*. They should be the random part of the data. If the residuals can be predicted in some way, we are leaving some predictive power on the table, and we can probably improve our model.
- *Be normally distributed, with constant variance*. The raw residuals are not expected to be normal. However, as deviance residuals have been adjusted to the shape of the underlying distribution, they are expected to be normal and with constant variance (***homoscedasticity***). If not, the selected distribution is possibly incorrect.

There are two ways we can assess the normality of deviance residuals for a model of claim severity built using the gamma distribution.



Firstly, a histogram of the deviance residuals, with the best normal curve fit super-imposed. If the random component of the outcome indeed follows a gamma distribution, we would expect the histogram and the normal curve to be closely aligned. In the case above, however, the histogram appears right-skewed relative to the normal curve, which suggests that the data exhibits greater skewness than what would be captured by a gamma distribution.

Another way to compare is the q-q plot. Here, the theoretical normal quantile for each point is placed on the x-axis, and the empirical (sample) quantile of the deviance residual is plotted on the y-axis. If the deviance residuals are indeed normal, the points should follow a straight line. In the case above, the edges of the distribution lie above the line, particularly the right-most points, which means that there are many more high-valued deviance residuals than would be expected under a normal distribution. So, the data is more skewed than gamma. An inverse Gaussian distribution, which assumes greater skewness, may be more appropriate for this data. This is shown below.

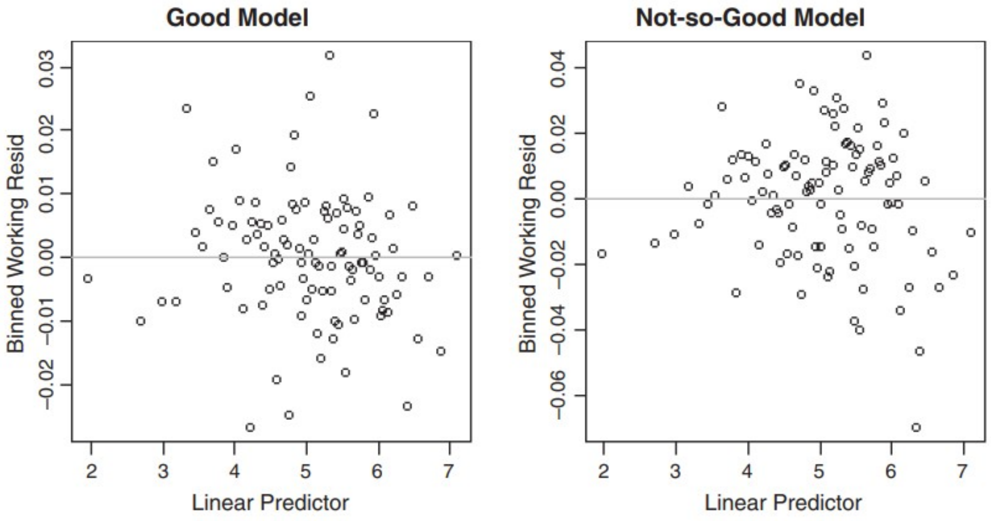


Discrete Distributions

For discrete distributions (Poisson, negative binomial) or distributions that otherwise have a point mass (Tweedie – a point mass at zero), the deviance residuals will likely not follow a normal distribution. Deviance residuals are less useful here. A possible solution here is to use *randomized quantile residuals*. They have similar properties to deviance residuals, but with added random jitter to wind up more smoothly distributed over the distribution.

Plotting Residuals over the Linear Predictor

Plotting residuals over the value of the linear predictor may reveal “miscalibrations” in the model – areas of the prediction space where the model may be systematically under- or over-predicting. Below shows binned working residual plots; the underlying models have thousands of observations but the working residuals have been binned into 100 bins prior to plotting.



The left-hand plot reveals no structural flaw in the model. The plot points form an uninformative cloud with no apparent pattern, as they should for a well-fit model. The right-hand plot is flawed; the residuals in the left region tend to be below the zero line (the model predictions for those observations are higher than they should be). The model then under-predicts in the middle region, and finally over-predict again in the right-hand region. This may be caused by a non-linear effect that may have been missed. The issue may be made clearer with plots of residuals over the various predictors.

Assessing Model Stability

Model stability refers to the sensitivity of a model to changes in the modelling data. We assume that past experience will be a good predictor of future events, but small changes in the past that we’ve observed should not lead to large changes in the future we predict.

- For example, if we experience an unusually large loss by an insured in a class with few members. A model run on all of the data may tell us that class is very risky. But if we remove the large loss from the dataset, the model may tell us that the class is very safe. The model is not stable with respect to the indication for this class, so we may not want to give full weight to its results.

Influential records tend to be highly weighted outliers. Assessing the impact of influential records is a straightforward way to assess model stability. A common way to do so is with ***Cook's distance***. Sorting records of Cook's distance will identify those that have the most influence on the model results – a higher distance indicates a higher level of influence. If rerunning the model without some of the most influential records in the dataset causes large changes in some of the parameter estimates, we may want to consider whether or not those records or the parameter estimates they affect should be given full weight.

Another way to assess model stability is with cross validation. This means testing the out-of-sample- model performance. The model should produce similar results when run on separate subsets of the initial modelling dataset.

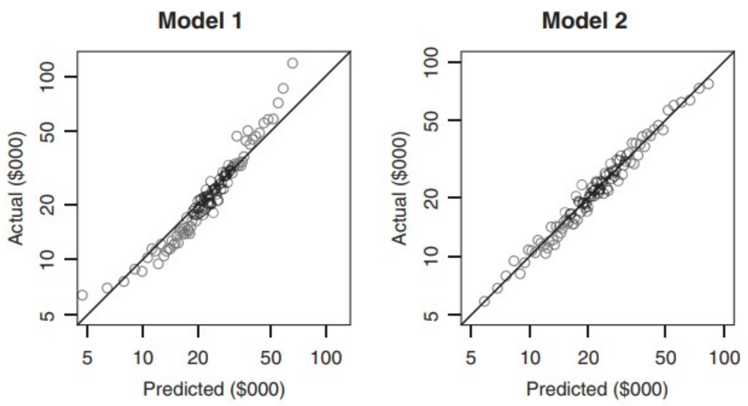
Still another way is via ***bootstrapping***. Here, the dataset is randomly sampled with replacement to create a new dataset with the same number of records as the initial dataset. By refitting the model on many bootstrapped versions of the initial dataset, we can get a sense of how stable each of the parameter estimates are. In fact, we can calculate mean, variance and confidence intervals via bootstrapping.

6 Model Validation and Selection

The process of model refinement is really a process of creating two candidate models and comparing them. All model refinement involves is model selection. But sometimes, model selection can be used for goals other than model refinement, such as choosing between alternate final models. This is often a business decision, which often has little to do with technical jargon.

Assessing Fit with Plots of Actual vs Predicted

A simple and easily understandable diagnostic to assess and compare the performance of competing models is to create a plot of the actual target variable (on the y-axis) versus the predicted target variable (on the x-axis) for each model. If a model fits well, then the actual and predicted target variables should follow each other closely.



Clearly, model 2 fits the data better than model 1. However, three cautions before using these plots:

- Create these plots on the holdout data. If created on the training data, they would look fantastic due to overfitting but in reality they might have little predictive power.
- Aggregate the data before plotting, due to the size of the dataset. A common approach is to group the data into percentiles. Sort the data by the predicted target variable, group it into 100 buckets, such that each bucket has the same aggregate model weight. The averages of the actual and predicted targets within each bucket are calculated and plotted as above.
- It is often necessary to plot the graph on a log scale, as done above. Without this, the plots would not look meaningful, since a few large values would skew the picture.

Measuring Lift

Model lift is the economic value of a model. This doesn't necessarily mean the profit an insurer can expect to earn as a result of implementing a model, but rather refers to a model's ability to prevent adverse selection. The lift measures attempt to visually demonstrate or quantify a model's ability to charge each insured an actuarially fair rate, thereby minimizing the potential for adverse selection.

Model lift is a relative concept, so it requires two or more competing models. It doesn't make sense to talk about the lift of a specific model, but the lift of one model over another. Again, this should be measured on the holdout set.

Simple Quantile Plots

Quantile plots are a straightforward visual representation of a model's ability to accurately differentiate between the best and the worst risks. Assume there are two models, A and B, both of which produce an estimate of the expected loss cost for each policyholder. Quantile plots are created thus:

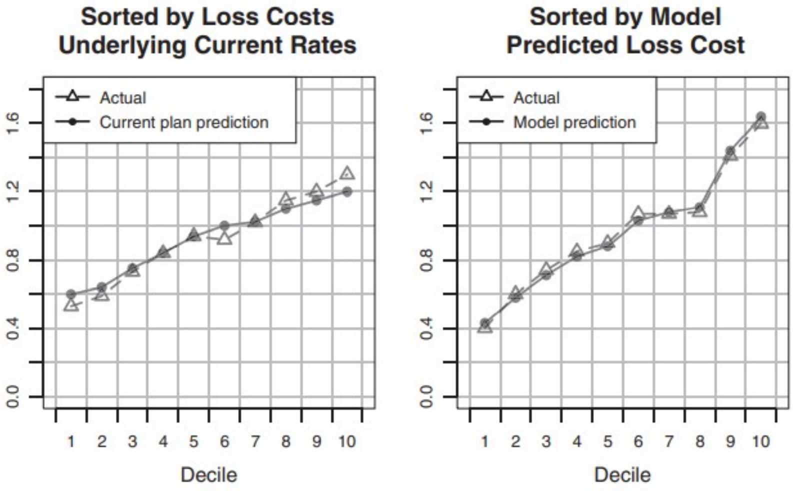
1 Sort the dataset based on the Model A predicted loss cost (smallest to largest).
2 Bucket the data into quantiles, such that each quantile has the same volume of exposures. This can be quintiles (5 buckets), deciles (10), or vigintiles (20).

3. Within each bucket, calculate the average predicted pure premium (predicted loss per unit of exposure) based on the Model A predicted loss cost, and calculate the average actual pure premium.
4. Plot, for each quantile, the actual pure premium and the pure premium predicted by Model A.
5. Repeat steps 1 – 4 using the Model B predicted loss costs. There are now two quantile plots.
6. Compare the two quantile plots to determine which model provides better lift.

To determine the “winning” model, consider the following criteria:

- ***Predictive accuracy***. How well each model is able to predict the actual pure premium in each quantile.
- ***Monotonicity***. By definition, the predicted pure premium will monotonically increase as the quantile increase, but the actual pure premium should also increase (though small reversals are okay).
- ***Vertical distance between the first and last quantiles***. The first quantile contains the risks that the model believes will have the best experience, the last quantile the worst experience.

Below can be seen decile plots for an example comparison between the current rating plan (left) and a newly-constructed plan (right). In both plots, the solid line is the predicted loss cost (either by the current rating plan or by the new model) and the broken line is the actual loss cost. Which model is better?



- *Predictive accuracy*. For the right panel, the plotted loss costs correspond more closely between the two lines than for the left panel, indicating the new model seems to predict actual loss costs better than the current rating plan does.
- *Monotonicity*. The current plan has a reversal in the 6th decile, whereas the new model has no significant reversals.
- *Vertical distance between the first and last quantiles*. The spread of actual loss costs for the current plan is 0.55 – 1.30, so not very much. That is, the best risks have loss

costs that are 45% below the average, and the worst risks are only 30% worse than average. The spread of the proposed model is 0.40 to 1.60.

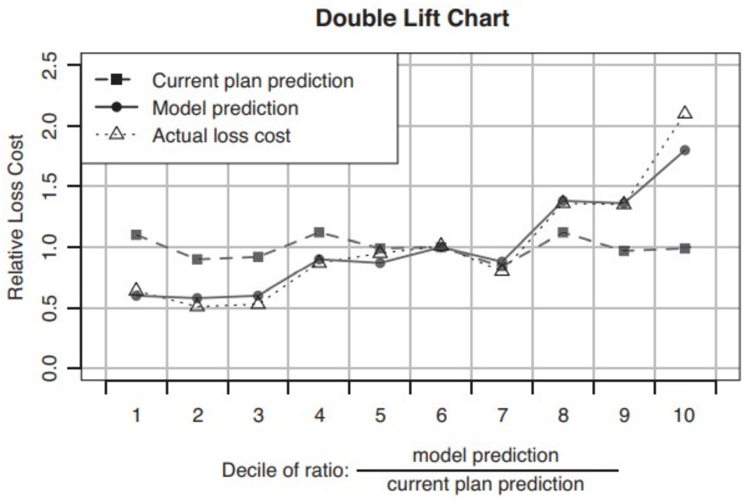
Thus, by all three metrics, the new plan outperforms the current one.

Double Lift Charts

Similar to the quantile plot, but it directly compares two models. Created via:

1. For each record, calculate Sort Ratio = (Model A Predicted Loss Cost)/(Model B Predicted Loss Cost).
2. Sort the dataset based on the Sort Ratio, from smallest to largest.
3. Bucket the data into quantiles.
4. Within each bucket, calculate the Model A average predicted pure premium, the same for B, and the actual average pure premium. For each of those quantities, divide the quantile average by the overall average.
5. For each quantile, plot the three quantities calculated in the step above.

In a quantile plot, the first quantile contains those risks which Model A thinks are best. In a double lift chart, the first quantile contains those risks which Model A thinks are best *relative to Model B*. In other words, the first and last quantiles contain those risks on which models A and B disagree the most (in % terms). The “winning” model is the one that more closely matches the actual pure premium in each quantile.



Above, it is clear that the proposed model more accurately predicts actual pure premium by decile than does the current rating plan. Specifically, consider the first decile. It contains the risks that the model thinks are best relative to the current plan. As it turns out, the model is correct.

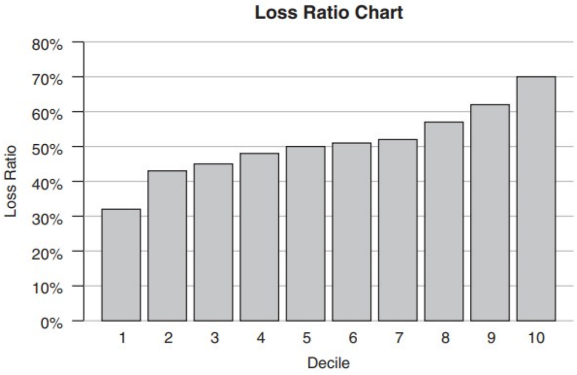
Loss Ratio Charts

Rather than plotting the pure premium for each bucket, the loss ratio is instead plotted. The steps for this are similar to that for creating a quantile plot:

- Sort the data based on the predicted loss ratio (=[predicted loss cost]/premium).

- Bucket the data into quantiles, such that each quantile has the same volume of exposures.
- Within each bucket, calculate the *actual loss ratio* for risks within that bucket.

Ideally, the model is able to identify deficiencies in the current rating program by segmenting the risks based on loss ratio. If a rating plan is perfect, then all risks should have the same loss ratio. That the model is able to segment the data into lower and higher loss ratio buckets is a strong indicator that it is outperforming the current rating plan.



The advantage of loss ratio charts over quantile plots and double lift charts is that they are simple to understand and explain. Loss ratios are a common metric in determining insurance profitability, so stakeholders should understand these plots.

Validation of Logistic Regression Models

Receiver Operating Characteristics (ROC) Curves

For logistic regression models, the GLM yields a prediction of the probability of the occurrence of the modelled event. Many of the previously discussed model validation diagnostics would work here too. For such models, a ***ROC curve*** is commonly used due to its direct relation to how such models are often used in practice.

For many practical applications, the probability of the event occurring will need to be translated into a binary prediction, for the purpose of deciding whether to take a specific action in response. We can make such a determination by choosing a specific probability level, called the ***discrimination threshold*** – say, 50% - above which we take action (e.g. investigate the potential fraud claim) and below which we do not. This determination may be thought of as the model's "prediction" in a binary sense. Under this arrangement, four outcomes are possible:

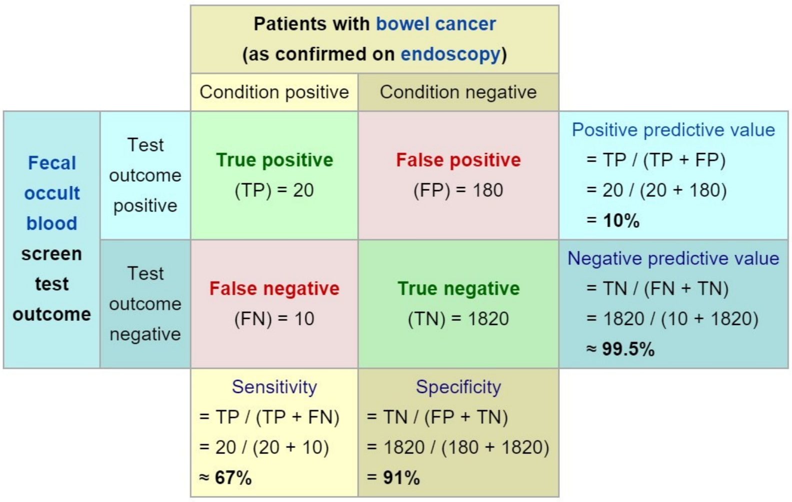
- The model predicts that the claim contains fraud ($\mu_i > 0.50$), and the claim is indeed found to contain fraud. This outcome is called a *true positive*.
- The model predicts fraud, but the claim does not contain fraud (i.e. a *false positive*).
- The model predicts no fraud ($\mu_i < 0.50$), but the claim contains fraud (i.e. a *false negative*).
- The model predicts no fraud, and the claim does not contain fraud (i.e. a *true negative*).

Outcomes #1 and #4 are successes of the model. #2 and #3 are its failures and each comes with a cost. The false negative allows a fraudulent claim to go undetected, resulting in unnecessary payment. The false positive also incurs a cost in the form of unnecessary resources expended on a claims investigation as well as possible impairment of goodwill with the insured.

A 100% prediction accuracy is unlikely in real-life, so false negatives and false positives are real possibilities. The selection of the discrimination threshold involves a trade-off: a lower threshold will result in more true positives and fewer false negatives, but at the cost of more false positives and fewer true negatives.

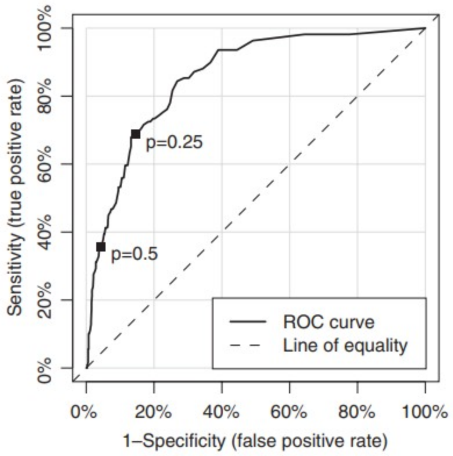
We can assess the relative likelihoods of the four outcomes for a given model and for a specified discrimination threshold using a test set. We use the model to score a predicted probability for each test record, and then convert the predictions of probability into binary predictions using the discrimination threshold. We then group the records by the four combinations of actual and predicted outcomes, and count the number of records falling into each group. We display the results in a 2×2 table called a *confusion matrix*.

The ratio of true positives to total positive events is called the ***sensitivity***. In the example below, the value is 20/30 = 0.67. With a threshold of 50%, we can expect to catch 67% of all fraud cases. The ratio of true negatives to total negative events is called the ***specificity***, and is 1820/2000 = 0.91 below. The complement of this ratio, called the *false positive rate*, is 1 – 0.91 = 0.09. This indicates that the hit rate of 67% comes at the cost of also needing to investigate 9% of all non-fraud claims. Lowering or raising the threshold would change these results.



		Patients with bowel cancer (as confirmed on endoscopy)		
		Condition positive	Condition negative	
Fecal occult blood screen test outcome	Test outcome positive	**True positive** (TP) = 20	**False positive** (FP) = 180	Positive predictive value = TP / (TP + FP) = 20 / (20 + 180) = **10%**
	Test outcome negative	**False negative** (FN) = 10	**True negative** (TN) = 1820	Negative predictive value = TN / (FN + TN) = 1820 / (10 + 1820) ≈ **99.5%**
		Sensitivity = TP / (TP + FN) = 20 / (20 + 10) ≈ **67%**	Specificity = TN / (FP + TN) = 1820 / (180 + 1820) = **91%**	

A graphical tool for evaluating the range of threshold options available to us for any given model is the ROC curve, which is constructed by plotting the false positive rates along the x-axis and the true positive rates along the y-axis for different threshold values along the range [0,1].



Above shows an ROC curve. The point (0,0) represents a threshold of 100% with which we catch no fraudulent claims (but investigate no legitimate claims either). Moving rightward, we see that lowering the threshold and incurring some false positives yields large gains in the hit rate; however, those gains eventually diminish for higher false positive rates.

The ROC curve allows us to select a threshold we are comfortable with after weighing the benefits of true positives against the cost of false positives. Determination of the optimal threshold is typically a business decision. The level of accuracy of the model will affect the severity of the trade-off. A model that yields predictions that are no better than random will yield true and false positives in the same proportions as the overall mix of positives and negatives in the data, regardless of the threshold chosen. For such a model, the ROC curve will follow the line of equality. A model with true predictive power will yield true positives at a higher rate than false positives, resulting in a ROC curve that is higher than the line of equality. Improved accuracy of the model will move the ROC curve closer to the top left corner of the graph, indicating that the model allows us a better hit rate for any level of false positive cost (as any threshold below 100% would correctly identify all fraud cases and trigger no false positives).

The model accuracy as indicated by the ROC curve can be summarized by taking the area under the curve, ***AUC***. A model with no predictive power will yield an AUC of 0.50. A “perfect” model would have an AUC of 1.0. The AUC measure bears a direct relationship to the Gini index and so they should not be taken as separate validation metrics, since an improvement of one will automatically yield an improvement in the other.

7 Model Documentation

Model documentation serves at least three purposes:

- To serve as a check on your own work and to improve your communication skills.
- To facilitate the transfer of knowledge to the next owner of the model.
- To comply with the demands of internal and external stakeholders.

Check Yourself

You’re going to make mistakes or overlook something. The better you are at identifying and correcting mistakes you’ve made in your own work, the easier life will be. How to find mistakes in your own work? One of the best ways is to *try to explain what you’ve done in writing*. When you write down what you’ve done in a way that allows someone else to understand it, you’re forced to revisit your work in full detail and to think critically about the decisions you’ve made along the way. This can bring errors to the surface. This is also true when you share your documentation with others; it may be easier for a peer to identify a conceptual error in a narrative than to detect it in a package of code.

Another benefit of documentation is that it serves to reinforce your understanding of the work. “To teach is to learn twice over”. The level of understanding required to document or explain a topic is greater than that required to simply execute. When you start from a foundation of deeper understanding, your subsequent work product will be of higher quality and will better stand up to scrutiny. *Document your work as you go.*

Stakeholder Management

Models need to be maintained and rebuilt. This task may fall to someone else or even to you. In either case, good documentation make’s everyone’s lives easier. Furthermore, others (regulators) may develop an interest in the models that you build. Your documentation should include:

- Everything needed to reproduce the model from source data to model output.
- All assumptions and justification for all decisions.
- Data issues encountered and their resolution.
- Any reliance on external models or external stakeholders.
- Model performance, structure, shortcomings.

8 Variations on the Generalized Linear Model

The GLM is a flexible, robust, and highly interpretable model that can accommodate many different types of target variables and covariate relationships. However, it does have a number of shortcomings:

- Predictions must be based on a linear function of the predictors. There are workarounds to handle non-linearity (such as polynomials), but those must be explicitly specified by the modeler.
- GLMs exhibit instability in the face of thin data or highly correlated predictors.
- Full credibility is given to the data for each coefficient, with no regard to the thinness on which it is based.
- GLMs assume the random component of the outcome is uncorrelated among risks.
- The exponential family parameter Φ must be held constant across risks.

Many of the more advanced predictive modelling techniques such as gradient boosting machines and neural nets do not have these flaws, and are therefore able to produce stronger models that yield more accurate predictions. However, these methods often entail a huge loss of interpretability, which for many actuarial applications is as great a necessity as predictive accuracy.

A number of extensions to GLMs have been developed that address some of the limitations above. As each of the following models is either based on the GLM framework or something similar, using them sacrifices little or no loss in interpretability, while potentially yielding increased flexibility, robustness, and accuracy.

Generalized Additive Models (GAMs)

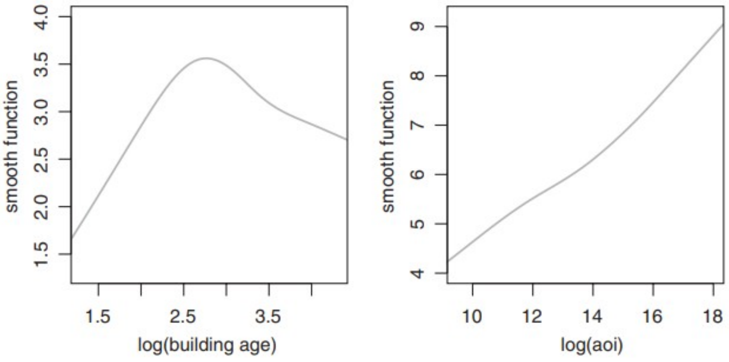
While non-linear effects can be accommodated by adding various transformations of the predictors into the linear equation, those are workarounds that must be specified manually. The ***generalized additive model*** (GAM) is a GLM-like model that handles non-linearity natively. Its specification is as follows:

$$y_i \quad Exponential(\mu_i, \Phi)$$

$$g(\mu_i) = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \ldots + f_p(x_{ip})$$

GAMs, like GLMs, assume the random component of the outcome to follow an exponential family distribution. The second equation has a twist: the addends making up the linear predictor are no longer linear functions of the predictors – they are any arbitrary functions of the predictors. Those functions, denoted f_1(.)…f_n(.) specify the effects of the predictors on the (transformed) mean response as smooth curves.

Note that the “additive” in the name refers to the fact that the linear predictor is a series of additive terms (though free from the constraint of linearity). As with a GLM, we can specify a log link, which would turn the model multiplicative. Unlike in a GLM, where the effect of a variable on the response can be easily determined by examining its coefficient, for GAM we are provided no such convenient numeric description of the effect. Predictor effects must be assessed graphically.



For building age, the GAM estimated a non-linear function, with mean severity first rising, reaching a peak at around building age $e^{2.8}$ = 16 years, then declining. For amount of insurance, although the GAM was free to fit any arbitrary function, the one it estimated was nearly linear.