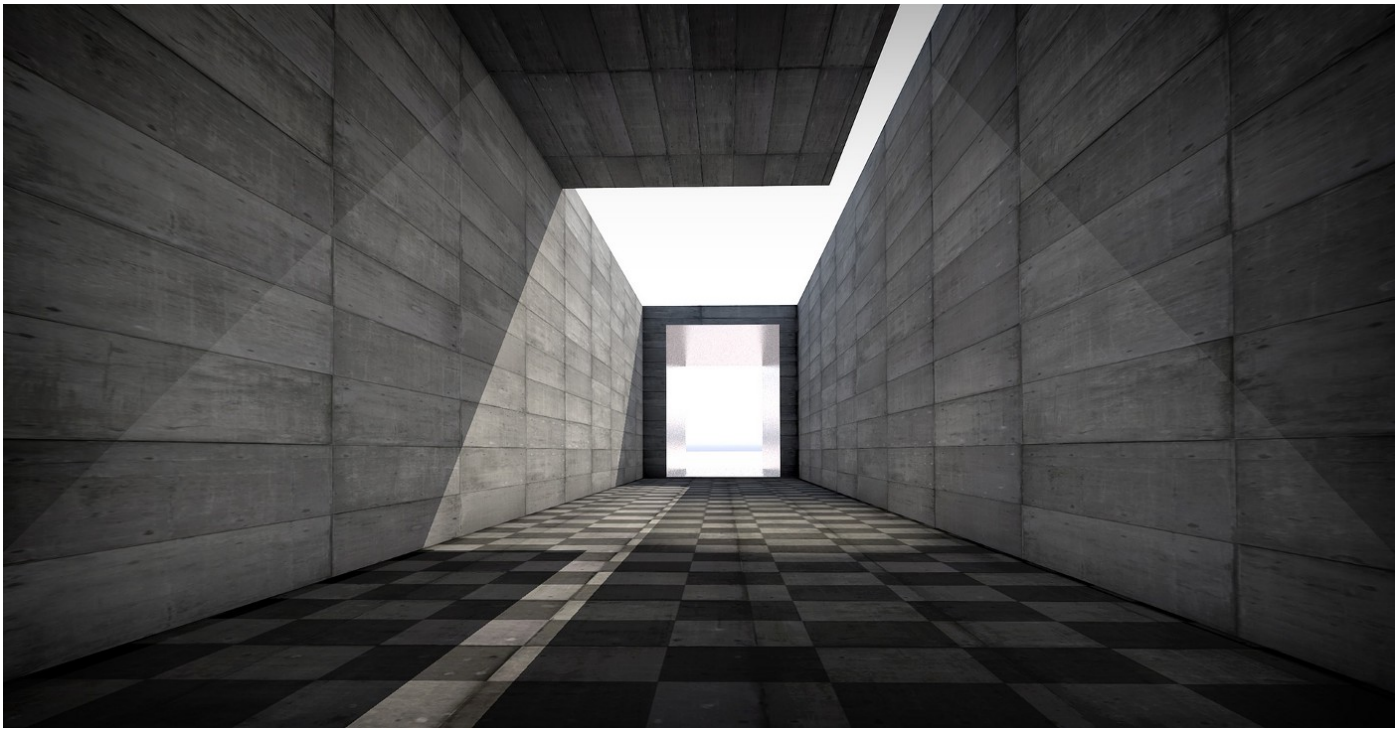
Beyond Linear Regression: An Introduction to GLMs

Genevieve Hayes



Coming from a statistics background, my first foray into data science and machine learning was via linear regression. At the time, I genuinely believed there was no statistical modelling problem so complex it couldn't be solved using a linear regression model that was appropriately defined.

At that same time, I also believed that a dataset containing 5000 data points was "big"; that learning SAS was more valuable to my career than learning R; and that *Buffy the Vampire Slayer* was the greatest television show ever made.

I was an undergraduate and my world view was extremely narrow.

Yet, in the universe defined by my second-year undergraduate statistics

class, I was actually right. There really *was* no problem so complex that it couldn't be modelled using a linear regression model. Any problem that was too complex for a linear regression modelling was conveniently hidden from us, so for all practical purposes, didn't exist within that world.

This highlights the trap a lot of data scientists, particularly those early in their careers, risk falling into.

A lot of introductory machine learning courses only teach students a certain finite set of possible models (usually linear and logistic regression, decision trees, naïve Bayes, support vector machines and neural networks) and only provide examples using datasets where one or more of these models is appropriate.

This can lead data scientists to the mistaken belief that all supervised learning problems can be solved with one of a small set of machine learning models.

But the models taught in introductory machine learning courses are certainly not the only statistical or machine learning models out there.

One such model, which is rarely taught in machine learning MOOCs or university Data Science degrees, is the generalized linear model or GLM. GLMs are frequently used in insurance premium setting and have proven to be one of the most useful statistical models I have encountered in my career to date.

In this article, we take a closer look at this highly versatile, yet too often ignored, model.

Linear Regression Recap

A (multiple) linear regression model is made up of three components:

1. An output (or response/dependent) variable, Y, where all observations of this variable are assumed to be independently drawn from a normal distribution with constant variance, s2;
2. A vector of k input (or explanatory/independent) variables, X1, X2, ..., Xk; and
3. A vector of k+1 parameters, b0, b1,..., bk, which allow us to express the mean of Y as a linear combination of our input variables:

Together, these three components define the probability distribution of Y as Yi ~ iid N(mi, s2), and our aim in fitting a linear regression model is to determine the parameter values that will optimally define that distribution for our data.

This can be done using several different methods, including ordinary least squares (typically used for smaller datasets) and gradient descent (typically used for larger datasets).

The key strengths of linear regression are that linear regression models are:

- fast to train and query;
- not prone to overfitting and make efficient use of data, so can be applied to relatively small datasets; and
- are easy to explain, even to people from a non-technical background.

However, a key weakness of linear regression is the restrictive assumptions that underlie it, which must hold for the model to provide a good fit to the data. In particular, the normality assumption.

By definition, normally distributed data is continuous, symmetrical and defined over the entire number line. That means that any data which is

either discrete, asymmetrical, or can take on values only within a limited range, really shouldn't be modelled using a linear regression.

As anyone who's taken undergraduate statistics will know, there are certain work arounds you can use if your data doesn't exhibit one or more of these characteristics. For example, if our data is skewed, we can transform it by taking the log or square root of our outputs.

However, this can be seen as a classic case of fitting our data to our model, which is never the best solution to a problem. An alternative approach is to use a different type of regression model, which is specifically designed for use with non-normal data. This is where generalized linear models come in.

Introducing Generalized Linear Models

Generalized linear models (GLMs) can be thought of as a generalization of the multiple linear regression model. GLMs are also made up of three components, which are similar to the components of a linear regression model, but slightly different. Specifically, GLMs are made up of:

1. An output variable, Y, where all observations of this variable are assumed to be independently drawn **from an exponential family distribution**;
2. A vector of k input variables, X1, X2, ..., Xk; and
3. A vector of k+1 parameters, b0, b1,..., bk, **and a link function g()**, which allow us to write **g(E(Y))** as a linear combination of our input variables. That is:

where m = E(Y).

Again, our goal is to determine the optimal parameter values to define the probability distribution of Y, but we are now no longer constrained to Y only

following a normal distribution.

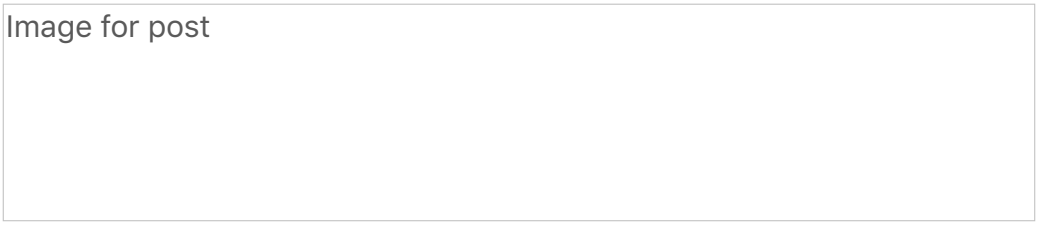
Under the GLM assumptions, Y can now follow any probability distribution within the "exponential family", which includes not only the exponential distribution, but also the normal, gamma, chi-squared, Poisson, binomial (for a fixed number of trails), negative binomial (for a fixed number of failures), beta and lognormal distributions, among others.

Pretty much every probability distribution that is commonly taught in undergraduate and Masters level statistics courses is a member of the exponential family.

The purpose of our link function is to transform our output variable, so that we can express it as a linear combination of our input variables (it is, NOT, to transform our output variable to normality, as is often mistakenly believed to be the case).

Depending on the probability distribution from which we assume our output distribution to be drawn, there are certain link functions that are commonly used.

For example:



When specifying a GLM, it is therefore, necessary to specify the output probability distribution function and the link function.

A standard linear regression model is a special case of a GLM where we assume a normal probability distribution and an identity link.

Three Situations Where GLMs Are Better Than Linear Regression

GLMs typically outperform linear regression models in cases where the normality assumption is violated. Three situations in which this can occur are the cases of: count data; skewed data; and binary data.

Let's look at how GLMs can be used in each of these situations.

Case 1: Count Data

Count data is any data that can only take on non-negative integer values. As the name suggests, it typically arises when counting observations of a particular type of event over a set period of time. For example, the number of car accidents at an intersection per year; or the number of times a process fails each day.

In order to fit a GLM to count data, we need to assume a probability distribution and link function for our model. Since count data is discrete, the probability distribution must also be discrete.

The most common choice in this situation is a Poisson or negative binomial distribution, with a log link function.

To fit a Poisson or negative binomial GLM to our data, we can use Python's statsmodels package, using syntax similar to the following:

```
import pandas as pd
import statsmodels.api as sm count_model = sm.GLM(count_data['Y'
```

This syntax assumes our dataset is in the form of a Pandas dataframe called count_data, with output variable Y and input variables X1 and X2, and that we want to fit a Poisson GLM.

To fit a negative binomial GLM, instead of a Poisson GLM, all we need to do is change sm.families.Poisson to sm.families.NegativeBinomial.

Case 2: Skewed Data

Although asymmetric data can be skewed in either direction. In practice, most asymmetric data you will encounter is right or positive skewed, such as the dataset plotted below.

Image for post

As before, to fit a GLM to this data, we need to select a probability distribution and link function. Our probability function needs to be positively skewed, and the most common choice in such a situation is a gamma distribution with a log link.

To fit a gamma distribution with a log link to our data, using the statsmodels package, we can use the same syntax as for the Poisson GLM, but replace sm.families.Poisson with sm.families.Gamma

The gamma distribution is only defined for values greater than 0. Therefore, if our output variable Y can take on negative or zero, then it may be necessary to transform our data, by adding a sufficiently large constant to the output variable, prior to fitting the model.

For example, if our output variable could take on any non-negative value,

including zero, we could transform it by adding 0.01 to all Y values prior to fitting our model, shifting the minimum value to 0.01.

Case 3: Binary Data

As with the previous two examples, the abovedescribed GLM can be fitted using the statsmodels package, using the same syntax as before, but replacing sm.families.Poisson with sm.families.Binomial, and sm.genmod.families.links.log with sm.genmod.families.links.logit.

That is, if we assume our dataset is called binary_data, and our input and output variables are as previously defined, using the command:

```
binary_model = sm.GLM(binary_data['Y'], sm.add_constant(binary _
```

Alternatively, we could also fit this model using the Python scikit-learn package's sklearn.linear_model.LogisticRegression function.

To quote Shakespeare, "there are more things in heaven and earth, than are dreamt of in your philosophy," and there is a lot more to data science and machine learning than just the contents of your average introductory MOOC.

In this article, we introduced one type of model that many early career data scientists are unfamiliar with, but this is just the tip of the iceberg, and there are many more where that came from.

That isn't to say that every data scientist needs to know everything there is to know about every statistical or machine learning model in existence.

However, by just being aware of there being more to data science than just

the basic models, you are less likely to make the mistake of fitting the data to the model, when you really should be fitting the model to the data.