Predicting Claim Cost of Auto Insurance Claims

Introduction

In this report, we will address the methods used in predicting claim cost for the InsNova Auto Insurance Company. We are given various predictors to predict this and other supplemental information. A quick summary of all variables used in the analysis is listed below.

- `ID`: Policy key, identifier for the row
- `claim_cost`: The claim cost, the variable we are interested in predicting
- `claim_ind`: An indicator variable of whether a claim was filed or not, takes value 0 if was not a claim, 1 if there was a claim
- `claim_count`: The number of claims filed for a given ID
- `veh_value`: Market value of vehicle
- `veh_body`: Type of vehicle
- `veh_age`: Age of vehicle
- `gender`: Gender of driver
- `area`: Driving area of residence
- `dr_age`: Age of driver
- `exposure`: The amount of time a vehicle was "exposed" to potential accidents

Of the provided variables, `claim_cost`, `claim_ind`, and `claim_count` are only provided in the training dataset, so we can only use them in constructing the response variable, not for prediction of the claim cost.

In this report, we address which of the variables above proved to be useful across different methods in predicting the claim cost. We will address the best methods for predicting claim cost in the **Methods** section, followed by the benefits and drawbacks of each method considered in the **Discussion** section.

Framework

Prior to analyzing the dataset, we needed to do some exploratory data analysis and see the type of data we are provided with. We wanted to assess what transformations we can make on our response and the various predictors and possible useful interactions between the variables.

From here on, the analysis and building of predictive models follow the same general process. We split the training dataset into a training and validation set with a 70% training and 30% validation set. This was done to provide a method of testing our predictions and comparing to a known true value.

For each method described in the upcoming **Methods** section, we first constructed our model on the training set. We then predicted our model on the validation set. We then ran a normalized Gini index function to obtain the Gini index for the test values, a metric that describes how different our sorted predicted values are from the sorted actual values. By creating our best possible model, our Gini index would increase. We do not necessarily want to solely maximize the Gini index at the expense of the model predictions on claim cost.

Methods

Exploratory Analysis

To begin our analysis on this data, we want to understand the shape of the response and some of the important predictors. From a basic check, we found that only 7% of all observations included a claim being filed at all. This leads to a very skewed distribution of the claim cost variable, and we consider placing a logarithmic transformation on this variable to gain a sense of normality in the response. We do this below, adding 1 to the claim cost variable to allow for the log transformation to occur.

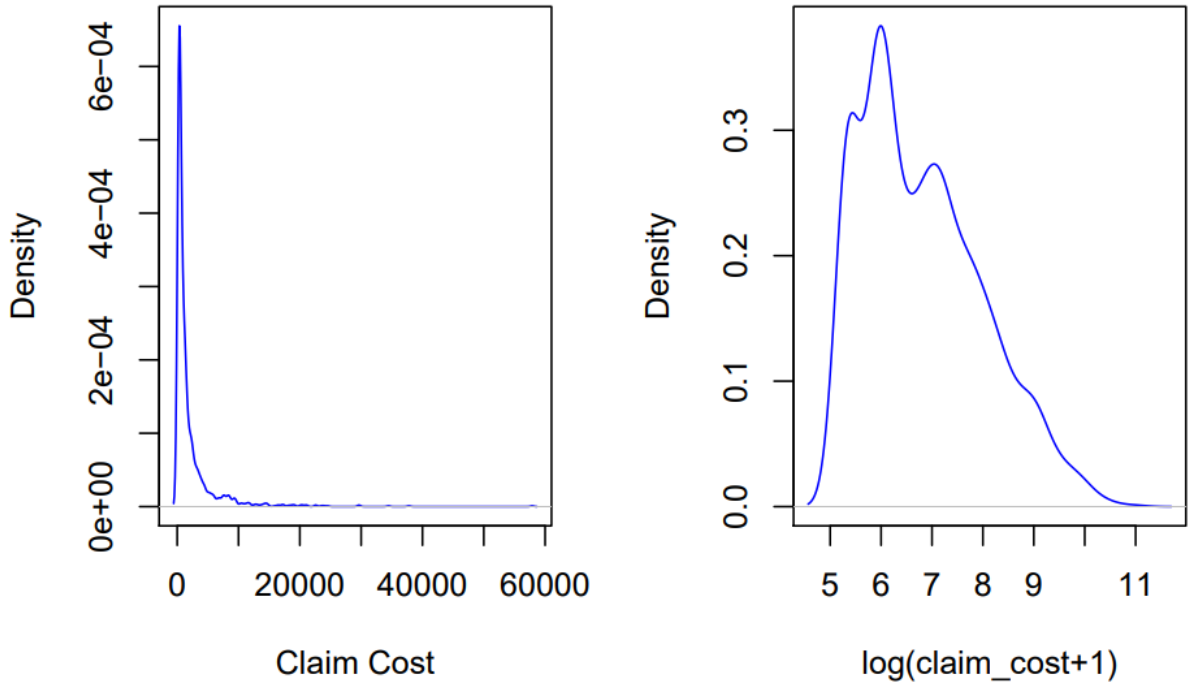

Figure 1: Log transformation of the response variable `claim_cost`

Above, we see that the response is slightly more normal, allowing for more flexibility in our model selection. We considered how the claim count could be a response variable. The boxplot describing the claim counts is below.

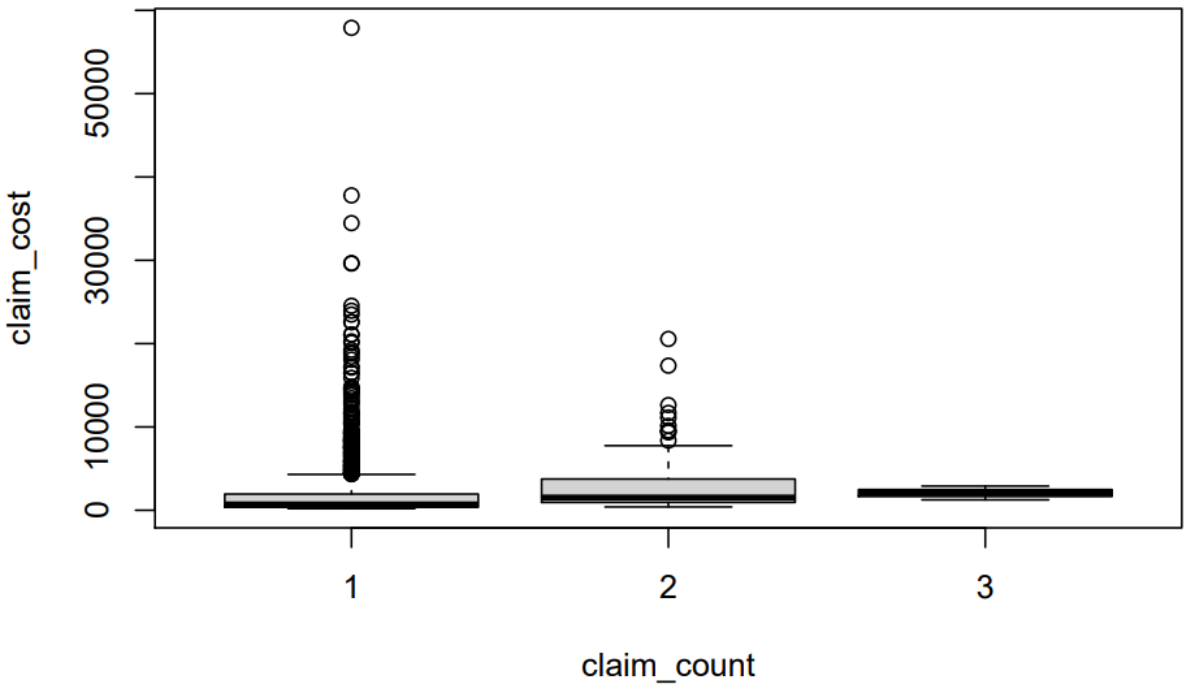

Figure 2: Boxplot visualization of `claim_count` against `claim_cost`

From the plot above, we see that relatively few observations have more than one claim filed. The vast majority of observations in the data have one claim filed, and the claim costs vary greatly among those. Thus, we decide to not include claim count in our model. In the next section, we discuss the methods and models we have attempted based on this exploratory analysis.

Model Selection

In our approach to model building, we wanted to focus on accurately predicting the claim cost given our available variables. We considered a few questions and tried to answer them appropriately. Primarily, we wanted to see how we can mitigate the influence of the observations where we did not file a claim, and work around that to build an appropriate model. Secondly, we considered the two-step approach, where we try to predict if a claim will be filed, then build a model on top of that. We will go through each of these ideas one by one and assess benefits and drawbacks of each model.

Linear Regression

Initially, we attempted a basic least-squares regression fit. We fit a model with all possible predictors, with `log(claim_cost+1)`. We did not initially include any interaction terms. This

model does not perform well and overfit the predictions, recording a low Gini score on multiple validation sets. However, this model is the most basic we could have made, so it was too soon to disqualify this method as viable. After this, we fit a linear regression model with all three way interaction terms, and performed backward stepwise selection with the AIC criterion. This model performed much better, and helped to reduce variance and did generalize well. However, this model would be difficult to interpret due to interaction terms and model complexity. We move on to explore other potential avenues that could provide better performance and reduce ambiguity about the model.

Two-step Model

Next, we attempted to predict the `claim_ind` variable first, then use those values to be weights on the predicted probability of `claim_cost`. To predict the probability of whether a claim would be filed, we would need to use classification methods such as logistic regression or random forest. However, when we attempted to do this, we found that there was no clear decision boundary between a claim existing or not existing. This means that modeling `claim_ind` would be nearly impossible, since there is a small subset of observations that have a claim at all and of those, there are not any strong associations between the given predictors and the probability of filing a claim. This method would be no better than randomly choosing weights to apply to `claim_cost`, so we moved on and did not pursue this method further.

Bootstrap Aggregation

The method that provided the best and most consistent results was the bootstrap aggregation method. We outline the steps taken below:

1. From our given training data, randomly sample 3000 rows with a claim (`claim_ind = 1`) and 3000 rows without a claim (`claim_ind = 0`).
2. Build a model on this new sampled dataset to predict the claim indicator or claim cost variable.
3. Repeat n times, average the predictions from the n models.

We built various models within this framework, including linear regression, logistic regression (to predict `claim_ind` once more), Tweedie GLM, and gradient boosted regression. The best model within the framework was the gradient boosted regression. We modeled `log(claim_cost+1)` as the response and followed a similar methodology to the linear regression previously. We fit all predictors first and saw how it performed, then fit all interaction terms and ran a backward stepwise selection to narrow down which variables are most useful in prediction.

The bootstrap method with gradient boosting proved to be the best in terms of Gini score and in variance reduction. It was also very straightforward to implement, and provided a lot of flexibility

in the model selection. The biggest benefit was the sampling in the bootstrap. We were not fitting on a largely skewed dataset where 93% of claim costs were zero, rather only 3000 of the 6000 in the dataset were zero. This allows the model to make more meaningful associations between what distinguishes a zero and nonzero claim cost, and in turn giving a better predictive performance for the Gini index. When it came to variance reduction, the inherent bootstrap sampling method and re-fitting the model n times allows for different models to be fit. Averaging the results prevents any one model's errors from dominating all of the predictions.

Discussion

When conducting this analysis, the main goal was to build a model with low error rates to provide good predictive performance for `claim_cost`. We used a variety of methods within the bootstrap framework described, the best of which was the gradient boosted regression. Overall, our results show that the bootstrap method helps reduce variance in the iterations of the gradient boosted regressions.

The next goal was determining which variables are most important in predicting claim cost. We created the following variable importance plot within the bootstrap framework of the gradient boosted model:

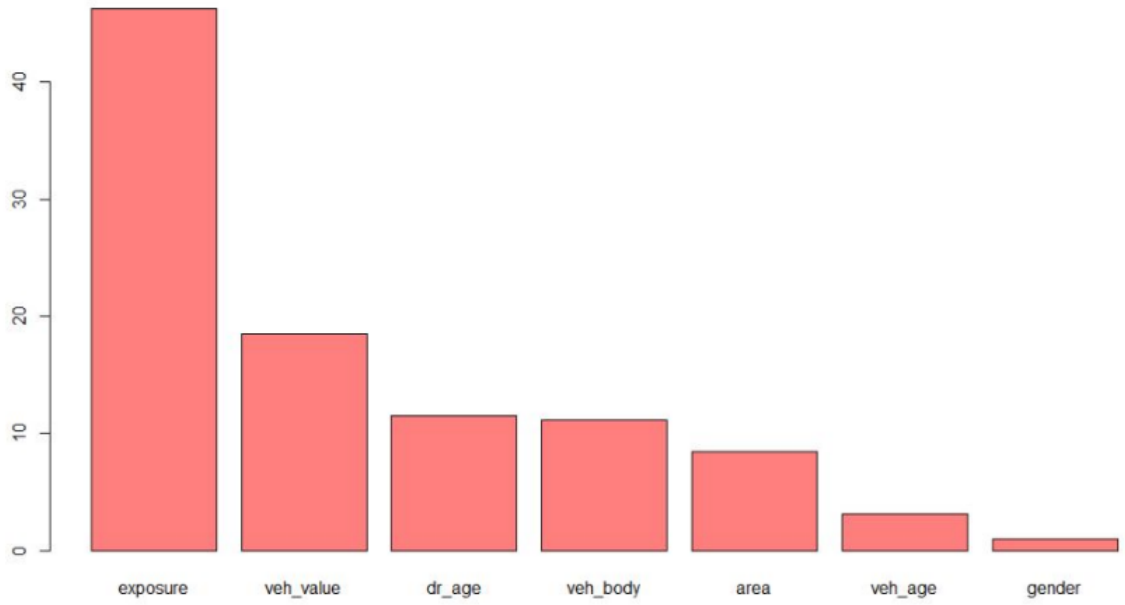

Figure 3: Relative influence of variables in bootstrap GBM

In the plot, we see that `exposure` and `veh_value` are most influential in predicting the claim cost for the policy. This is indicative of the fact that the cost is largely influenced by the risk factors associated with the car and the situation of the driver; the more time the vehicle is exposed to potential accidents, the higher the claim cost will be. A similar interpretation follows for the vehicle value as well.

In our model, we have some advantages and disadvantages. Perhaps the greatest benefit of the model is that the bootstrap methodology helps to reduce overfitting by providing a random selection of observations, and diversifying the selection of data to fit a model on, then ultimately averaging all of the fitted models. It is also easy to update the model in light of new regulations or real-world implications on the data. One of the key disadvantages for this model framework is the time it took for variable selection. With a dataset of this size, running multiple iterations of the gradient boosting algorithm can take time on local machines. It also does not account for any significant interactions between the variables, since that would increase the time it takes to fit the model and output predictions. There is also a bit of complexity in the regression. Gradient boosted methods are not very easy to understand, so the transparency of the model and adopting it widely without knowing what goes on behind the scenes can prove to be a challenge.

Further analysis can be conducted to further increase predictive performance as well. Some of the options considered were methods that could reduce variance and prevent overfitting even more, such as building a separate, non-bootstrapped model that performs well, like a Tweedie GLM, and averaging those predictions with the final bootstrapped predictions. This would be a good way to combine the two approaches and obtain a closer estimate of the claim cost and improve the usability of the model. Additionally, having more information in the dataset could be useful. For example, knowing the driver's past history would help since if the driver is more reckless in the past, their claim costs would be higher versus someone who was safer. It would also help to have a bit more insight on some of the provided variables. The variables `veh_age` and `dr_age`, corresponding to vehicle age and driver age respectively, are defined categorically, with 1 being the youngest and increasing thereon. We do not know what constitutes a "young" or "old" vehicle or car. If these were defined numerically with age in years, we could gain a better insight on how old the cars or drivers are rather than arbitrarily binning them into categories.

Overall, our results show that gradient boosting on a bootstrapped dataset performs best in predictive performance. We determined the most important predictors for our analysis and were able to adequately predict claim cost with this method.

References

Batzner, Kilian. "Intuitive Explanation of the Gini Coefficient." The Blog, 10 Oct. 2017, theblog.github.io/post/gini-coefficient-intuitive-explanation/.

Faraway, Julian James. Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models. Chapman amp; Hall/CRC, 2006.

"Gradient Boosting Machines." Gradient Boosting Machines · UC Business Analytics R Programming Guide, 2016, uc-r.github.io/gbm_regression.