**Reminder of project purpose (Aaron 1:00)**

Group 4, consisting of myself, Donald and Richard, are excited to share the results of our project with you. Our project consisted of taking British Motor Data from multiple years to predict accident severity. Response variable that we are using is categorical with three classes: Fatal, Serious, and Slight. These represent the severity of the Bodily Injury damage from the given accident. We will discuss specifics about our approaches later, but the uniqueness of our problem for us was to predict a variable with more than two classes. The proposed business use of the solution is to generalize the prediction past the specific location of the accident (not use UK specific location variables) and allow an insurance company to more accurately set insurance reserves after first notice of a claim.

**Data Sources (Richard 1:30)**

Originally, our idea was to merge as much data as we could. Through the UK website, we found over 40 files we intended to merge. These files spanned from 2005 to 2016 with 7 different types. After going through the files, we found many different issues with different file types. This included some files that were specifically for fatalities, some files without a key to match on and some files that had a one-to-many relationship. Due to these issues, we decided to solely focus on the accidents dataset and do some feature creation on data points we found interesting.

After merging each year of our data, from 2009 to 2016, we knew we needed a method to split our data into sets. We decided to hold out 2015 and 2016 for our final model. We read that there was a change in the way accidents were reported in 2016, so we held 2 years out in case 2016’s changes posed an issue with prediction. After removing 2015 & 2016 for our holdout, we partitioned the rest of our data into 70% for a training set and 30% for a testing data set.

**Data Cleaning and Variable Exploration (Richard 3:00)**

Add text here

**Feature Creation (Aaron 1:30)**

As was explained previously, we had many files that didn’t have a one to one relationship between the records. The accident level file was chosen as our exposure base for prediction, but we felt there were other variables from the vehicle and casualty file that might provide additional predictive lift.

The method for feature creation began with brainstorming what information we thought would provide additional predictive lift. We then looked through the data fields available in the other files to see if we could extend our brainstormed list. After we had compiled this list, we eliminated those not available and had to decide how to roll the available information into an accident level predictor. For example, we felt that children in the car might have a different probability of injury than adults. Age of the passengers IS present, but there are many passenger records for each accident. We decided to accomplish this by creating indicators. If any of the participants were under a certain age, we added a flag to that accident. We used this same method for other indicators, such as: Was there a motorcycle involved? Was one of the vehicles stationary at time of accident? Was one of the vehicles hit one the front/side/rear?

These indicator variables were created, summarized to the accident level, and merged into our main accident data frame.

**Generation of multiple models – RF and Penalized Multinomial Regression (Aaron – 1:30)**

We tried three different model types to predict severity: Random Forest, Penalized Multinomial Regression, and one-vs-rest logistic regression. The first two were available in the caret package, but the third required some additional thought...so we’ll spend some additional time on that one. Before we get to that, I’ll focus on the first two.

Since we had imbalanced predictors, it was necessary to downsample (through the caret package) to handle the three classes. Slight has roughly 85% of the exposure, with Fatal at only 2%. This meant that our initial model building attempts had everything predicting as Slight…with 85% accuracy! Woot!

After downsampling, we got numbers that were more reasonable (although less overall accuracy). For RF and Penalized Multinomial Regression (run through the multinom package) we ran 5 fold cross validation. Why 5? Because it didn’t make our computers blow up. The mtry for RF were selected as plus and minus one from the square root of the number of predictors. For Penalized Multinomial Regression, we left the default decay parameters.

Random Forest

1148417 samples

39 predictor

3 classes: 'Fatal', 'Serious', 'Slight'

No pre-processing

Resampling: Cross-Validated (5 fold)

Summary of sample sizes: 918734, 918734, 918734, 918733, 918733

Addtional sampling using down-sampling

Resampling results across tuning parameters:

mtry min.node.size Accuracy Kappa

5 10 0.5331948 0.1179604

5 20 0.5286956 0.1166777

6 10 0.5333080 0.1195131

6 20 0.5356983 0.1201219

7 10 0.5327298 0.1194116

7 20 0.5307576 0.1195539

Tuning parameter 'splitrule' was held constant at a value of gini

Accuracy was used to select the optimal model using the

largest value.

The final values used for the model were mtry = 6, splitrule

= gini and min.node.size = 20.

Cross-Validated (5 fold) Confusion Matrix

(entries are percentual average cell counts across resamples)

Reference

Prediction Fatal Serious Slight

Fatal 0.9 4.8 16.1

Serious 0.3 5.4 21.8

Slight 0.1 3.4 47.3

Accuracy (average) : 0.5357

Penalized Multinomial Regression

Cross-Validated (5 fold) Confusion Matrix

(entries are percentual average cell counts across resamples)

Reference

Prediction Fatal Serious Slight

Fatal 0.8 4.2 15.2

Serious 0.3 4.9 20.9

Slight 0.2 4.4 49.1

Accuracy (average) : 0.5484

You can see the confusion matrix and accuracy metrics on the screen.

**One vs Many classification approach (Donald 3:00)**

The last model we decided to run was multinomial logistic regression. It was completely new to us, but we devised a method to help us essentially do logistic regression on a variable with three ordinal categories. To begin, we knew we would need to create two separate logistic regression models. The first model we created was to predict fatal vs. non-fatal because, overall, we want to have the best accuracy on our fatal observations. The variables we used in this model were the same as we used in random forest and penalized multinomial regression. We also attempted to use StepAIC, but it didn’t give us any better of a prediction and had a few more variables. After fitting the model and running the prediction on our training data, we knew we had to choose a probability threshold that had a good prediction on fatal, not just the set as a whole. If we classified everything as slight, we’d get a prediction in the 90’s, so accuracy itself is not a good metric to use. The first plot is accuracy for sensitivity and the second plot is our ROC curve. We see from here that to balance between sensitivity and accuracy, we need to choose a point somewhere between 0.9 and 1. Since this felt a little arbitrary, we decided to use the coord function in the pROC library to optimize sensitivity mathematically. For fatal vs. non-fatal, we found that sensitivity was optimized with a probability cutoff of 0.9871327. This gave us an accuracy of about 73% and a sensitivity of about 72%.

Next, we created a model to predict serious vs. non-serious. Just as we did in the previous model, we use the same variables. After fitting our model and running prediction on the training, we get these two new plots. Looking at these new plots, we see that we need to choose a point somewhere between 0.8 and 0.9 to get an optimal choice that balanced accuracy and sensitivity. Again, we used the R function to calculate this for us and found our optimal probability threshold to be 0.8672438. This gave us an accuracy of close to 60% with a sensitivity of about 67%

This finalized our two smaller models. From here, we needed a way to put them together. To do this, we obtained the probability vectors for each model that we ran on the training set. We then created a vector that said if the probability for our fatal prediction was less than our optimized cutoff for fatal, then we’re identifying it as a fatal severity. Otherwise, if the probability for serious was less than our optimized cutoff for serious, then we’re identifying it as a serious severity. Anything left would be classified as a slight severity.

Running these predictions through a confusion matrix with the actual values gave us our final metrics for the training data. We played with several different probability cutoff points, essentially using this interactive confusion table below, but we decided to stick with our optimized sensitivity since it was calculated mathematically and seemed less arbitrary.

**Model Selection (Donald 2:00)**

The results from our confusion matrix are percentages of the overall number of observations. Since our models are all multinomial, there’s no clear-cut way to choose a model. First, looking at accuracy alone, each model is pretty close to each other, with the best and worst accuracy having a difference of only 4%, but as we said earlier, we can’t look at accuracy alone. This meant we would need to compare those categories that were the biggest concern to us, which was fatal and serious severities. In regards to fatal severities, we can see that logistic regression had a very slight edge, beating random forest by less than a tenth of a percentage. This led us to slightly favor random forest and logistic regression. Next, we looked at serious severities and random forest was the clear winner, with logistic regression having a relatively poor prediction in comparison to the others. Since random forest was the best prediction for serious and extremely close to the best for fatal, we decided that random forest would be the model to choose.

|  |  |  |  |
| --- | --- | --- | --- |
| RF on Test Data | Fatal | Serious | Slight |
| Fatal | 0.9% | 4.7% | 15.9% |
| Serious | 0.3% | 5.7% | 22.9% |
| Slight | 0.1% | 3.2% | 46.4% |
|  |  |  |  |
| Accuracy | 53.0% |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| MN on Test Data | Fatal | Serious | Slight |
| Fatal | 0.8% | 4.2% | 15.2% |
| Serious | 0.3% | 4.8% | 20.0% |
| Slight | 0.2% | 4.6% | 49.9% |
|  |  |  |  |
| Accuracy | 55.5% |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Logistic on Test Data | Fatal | Serious | Slight |
| Fatal | 0.9% | 6.0% | 20.9% |
| Serious | 0.2% | 3.8% | 17.5% |
| Slight | 0.2% | 3.9% | 46.8% |
|  |  |  |  |
| Accuracy | 51.5% |  |  |

Now, it was time to see how our final model performed on the holdout data, which were for the years 2015 and 2016. Running our final model on this data gave us an accuracy of 58.7%. Our prediction on fatal was slightly worse than on the training data, but our prediction on serious severities looked slightly better. We also ran them for the separate years, since there was a change in the way accidents were reported, but it didn’t make any significant difference in our predictions.

|  |  |  |  |
| --- | --- | --- | --- |
| MN on Holdout | Fatal | Serious | Slight |
| Fatal | 0.5% | 2.5% | 7.6% |
| Serious | 0.4% | 6.9% | 24.7% |
| Slight | 0.3% | 5.7% | 51.3% |
|  |  |  |  |
| Accuracy | 58.7% |  |  |

**Summary of what we learned? (Aaron 0:45)**

So…what did we learn during this? The biggest learning came from trying new methods to predict multi class responses. We grappled with the difficulties of how to pick the “best model” from a multi-class prediction. We gained additional experience dealing with large datasets on small machines. And we had fun creating new features, merging datasets, and exploring our data. If we would have had more time, we would have looked into an ensemble approach to putting the three methods together to see if we could get better results. We also would have extended our selected model into a simulated business case…to estimate business value. Overall, we enjoyed trying new things and working as a team.

Thank you.