Bingo Bonus – Logistic regression attempted. Random Forest feature importance was used in variable selection, which I think is the same as the decision tree bonus option.

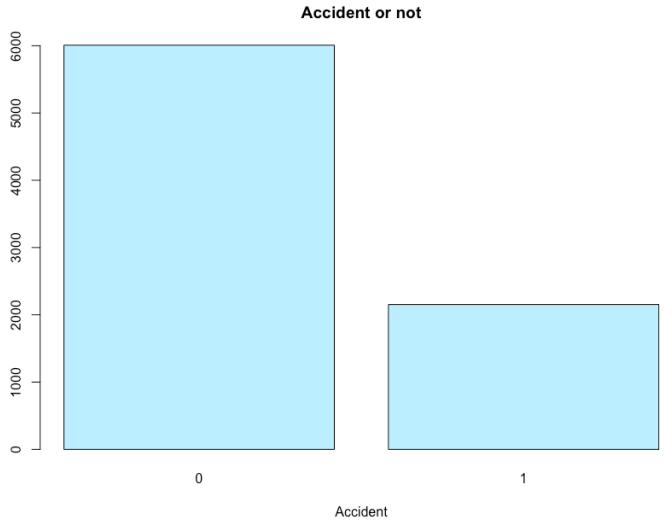
Introduction

There are two goals to this endeavor. The primary goal is to predict whether or not an insurance customer is going to have an accident. Secondarily, if they do have an accident, predict the amount the accident will cost the insurance company. The data we are working with have been split into test and training files to allow model testing to be done on out-of-sample data. The scoring portion of the requirements are at the end of the file. Because of how imputation was done, and how the selected model flows into the scoring code, a separate file was not really feasible. However, by changing the read-in-test-data statement, any arbitrary file can be scored.

Data Exploration

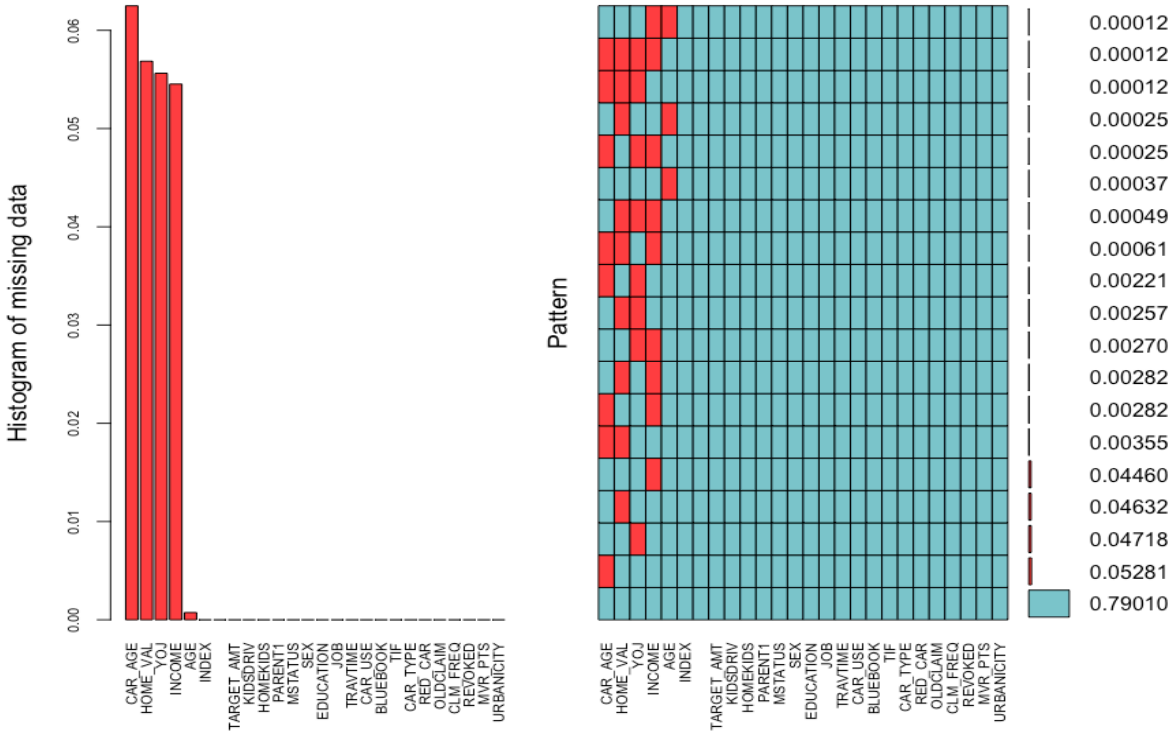
The data we are using have 2141 observations of 26 different variables (features). We are interested in predicting whether a driver will have an accident, and if they do, what it will cost. Figure 1 shows that most of the drivers do not have accidents.

Figure 1



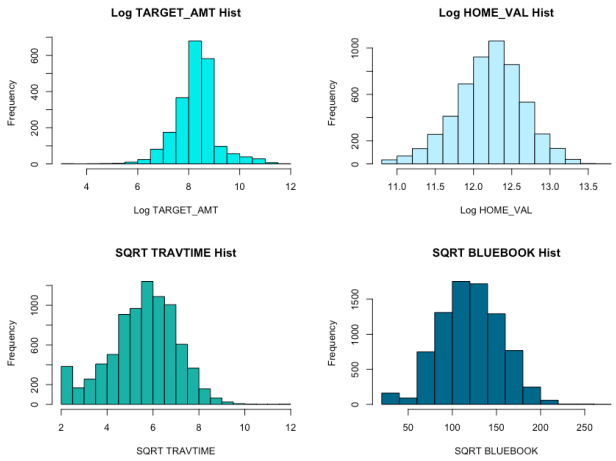
There is missing data to deal with. Figure 2 illustrates the number of missing values for each feature, and shows the frequency of co-occurring multiple missing values.

Figure 2



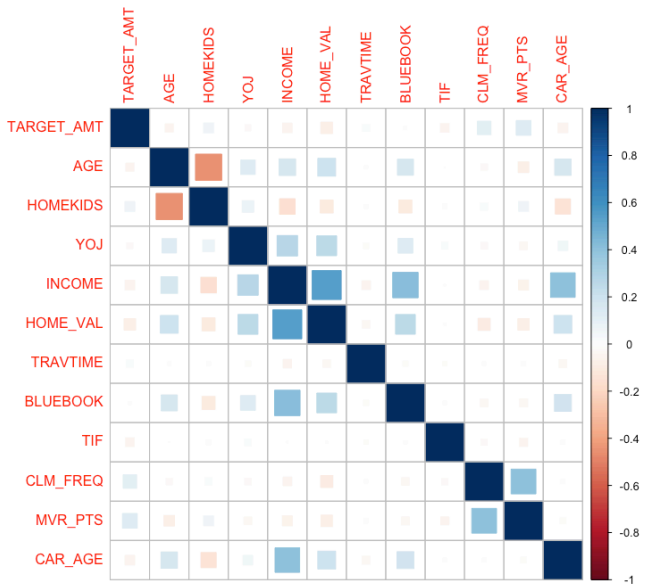
Five of the 26 variables have missing data which we will need to impute; 79% of all observations are complete. In several cases data needed transformation to be normally distributed. Once transformed, the data fit the assumed normal distribution; illustration the transformations is presented in Figure 3.

Figure 3



There are a few strong correlations in the data: age and having children at home, as well as the value of a home, age and bluebook value of the car. These correlations indicate there may be an issue with collinearity, the data may violate the assumption of variable independence.

Figure 4



Data Preparation

Preparation for this dataset required several different steps. After reading in the raw data, the roughly half of the features required mapping from vectors of values to factors for processing. Where present, "$" characters were removed from the factors as well. Flag-variables were created for the variables that required imputation. The missing data was imputed using the column mean for the missing values resulting in the Imp_df data frame.

Income data was binned into categories to explore the impact of grouping values on the modeling process. To normalize the data, the variables for travel time and bluebook values were transformed using square roots, and the home value and old claim values were transformed by taking their log. A new feature, "home owner" was added.

Finally, to gauge the use of binary values for the various factors, a new data frame using dummy variables (Sparse_df) was created. The SEX, EDUCATION, PARENT1, MSTATUS, REVOKED, URBANICITY, JOB, CAR_USE, CAR_TYPE, and INCOME_bin variables were converted to binary dummy variables. A 3rd data frame was created by removing columns from the dummy-data frame. Columns identified by the "findLinearCombos" command as having dependencies in Sparse_df data frame were deleted to create the data frame Short_sparse.

Model Construction and Selection

To get an idea about the relative importance of the different features, random forest models were fit and the feature importance was plotted for each of the 3 data frames.

Figure 5 - Regular data frame

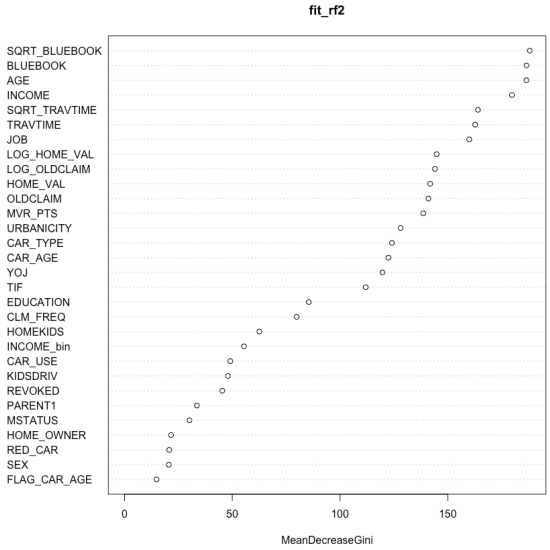


Figure 6 - Dummy Variable data frame

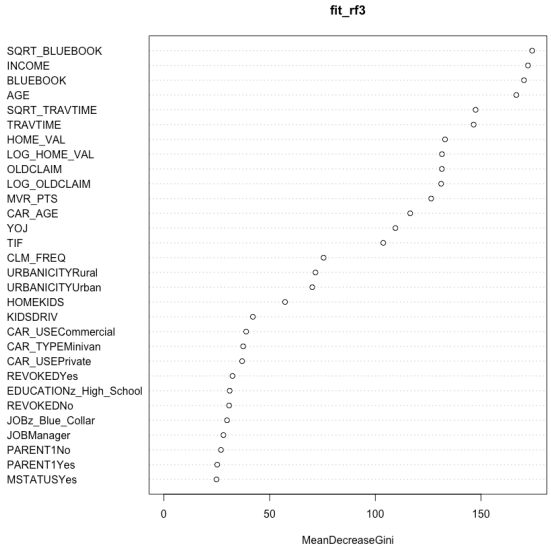
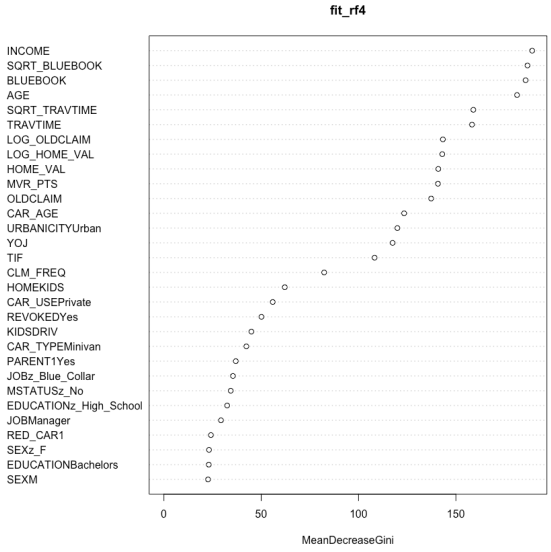


Figure 7 - Reduced dummy variable data frame



Importance was used to inform variable selection for the different models built. A total of 10 models were built - 2 using the Sparse_df data frame, 3 using the Short_sparse data, and 5 using the original data frame, Imp_df. Three of the attempted models failed to converge with the default number of iterations. In all 3 non-convergent cases, the model was built using all the variables in the dataset. Changing the models to use maxin = 50, did get convergent

models, with the warning “glm.fit: fitted probabilities numerically 0 or 1 occurred”. I concluded from this there is a collinearity problem, and removed those models from further consideration.

The results of the model evaluation are:

Table 1

Model #	1	2	3	4	5	6	7	8	9	10
AIC	N/A	N/A	7629.165	7378.261	7560.740	7352.801	7348.559	N/A	7401.023	7326.290
BIC	N/A	N/A	7832.371	7665.553	7770.954	7647.101	7621.837	N/A	7583.208	7676.646
Log Likelihood	N/A	N/A	7571.165	7296.261	7500.740	7268.801	7270.559	N/A	7349.023	7226.290
KS Statistic	N/A	N/A	0.4498	0.4681	0.4580	0.4719	0.4756	N/A	0.4630	0.4775

At first glance, Model10 looks like the best choice. It’s AIC, LogLikelihood, and KS statistic are the best of the models. However, looking at VIF for Model10, we find it has issues, there are aliased coefficients, so we have a multicollinearity problem there too. Model 7 looks like the next-best candidate, but both Models 6 and 7 have results which are good and the results are very close to each other. Model6 is built using the original data frame, while 7 uses the somewhat confusing reduced dummy variable data. Model6 also uses fewer variables.

The area under the curve (AUC) for the 3 models in contention is also similar. The rounded value of the AUC for both models 6 and 7 is 0.8156, Model10 has an AUC of 0.8182. The ROC curves for these 3 models are shown below.

Figure 8 - ROC model 6

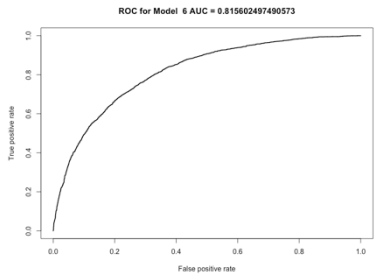


Figure 9 - ROC model 7

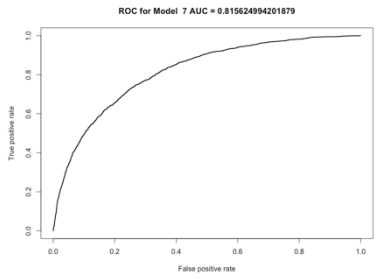
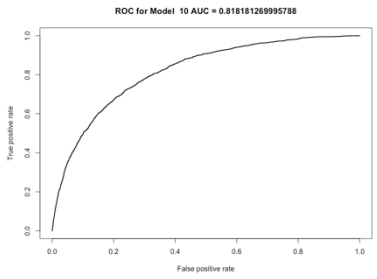


Figure 10 - ROC model 10



The Model6 coefficients are reported by R as being:

(Intercept), -3.3147385808,	**SQRT_BLUEBOOK,** -0.0057466627,	**SQRT_TRAVTIME,** 0.1686708784,	**AGE,** -0.0032295225,	**LOG_HOME_VAL,** -0.1714739167,	**LOG_OLDCLAIM** 0.0209935627,
JOBDoctor, -0.8064885702,	**JOBHome Maker,** -0.3346903116,	**JOBLawyer,** -0.2872872117,	**JOBManager,** -0.9743029465,	**JOBOther,** -0.4395407023,	**JOBProfessional** -0.2418529292,
JOBStudent, -0.3846058702,	**JOBz_Blue Collar,** -0.0984998711,	**MVR_PTS,** 0.1013792827,	**CAR_AGE,** -0.0002934738,	**YOJ,** 0.0167994745,	**CAR_TYPEPanel Truck,** 0.5678465666,
CAR_TYPEPickup, 0.5654716842,	**CAR_TYPESports Car,** 0.9940265884,	**CAR_TYPEVan,** 0.6502979720,	**CAR_TYPEz_SUV,** 0.7610922142,	**TIF,** -0.0544949370,	**URBANICITYUrb** 2.3858516919,
CLM_FREQ, 0.0895279541,	**EDUCATIONBachelors,** -0.3701511953,	**EDUCATIONMasters,** -0.2840217952,	**EDUCATIONPhD,** -0.2438989034,	**EDUCATIONz_High School,** 0.0215842899,	**HOMEKIDS,** 0.0257301077,
INCOME_binZero, 0.8274063253,	**INCOME_binLow,** 0.0854549698,	**INCOME_binMedium,** 0.0321849469,	**INCOME_binHigh,** -0.3425639131,	**KIDSDRIV,** 0.4078828120,	**CAR_USEPrivate** -0.7579796149,
REVOKEDYes, 0.7111026296,	**MSTATUSz_No,** 0.4777645208,	**PARENT1Yes,** 0.3906686707,	**HOME_OWNER,** 1.7368696887,	**RED_CAR1,** -0.0185424333,	**SEXz_F,** -0.0730184137,

Nothing in the coefficient list looks odd, at least not that I can identify. For example, more education contributes negatively to having an accident, having moving violation points contributes positively. This seems reasonable.

The residual checks for the model are:

Figure 11

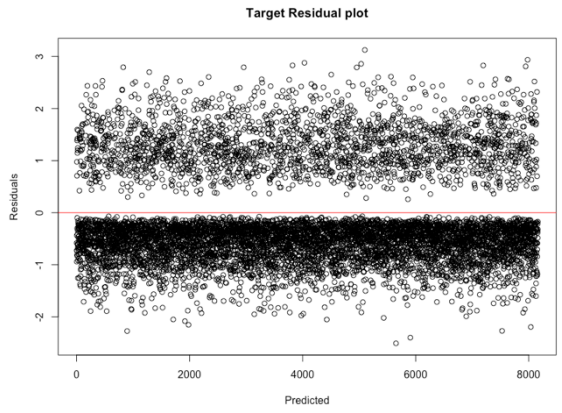
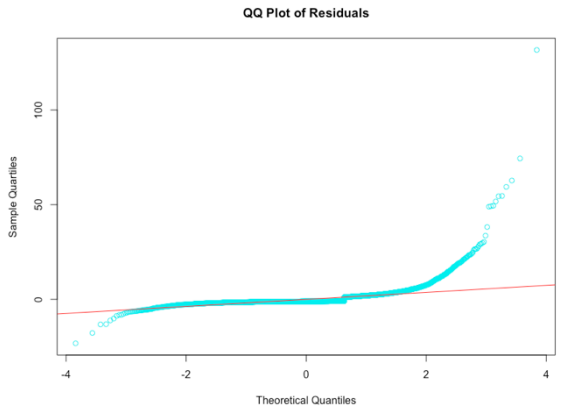


Figure 12



The QQ plot does deviate from normal at the ends, but I believe the residuals are well-enough distributed to be considered valid. The target plot shows the separation between the two prediction outcomes: likely and not likely to have a crash, so that seems fine to me, as the residuals look well distributed for each of the outcomes.

Final Recommendation

I prefer easier to understand models, which have minimal data prep requirements. So, I am selecting Model6 from the suite of choices. The AIC and BIC numbers are fairly close to the best values, as is the KS statistic and the Log Likelihood. The AUC value is the same as Model7 and only slightly less than Model10 the more complex, rejected models. The chosen model has the advantage of using the most straightforward of the data prep options, and has the least number of overall variables. It uses 23 features, compared to 30 for Model7 and 52 for Model10. The comparative simplicity of Model6, coupled with its strong performance on the test data make it my champion model.

The secondary goal of this effort is to predict the target amount. To do this, in the scoring section I set a cut-off value of .5. Any observation with a P_TARGET_FLAG great then .5 was assigned a P_TARGET_AMT calculated as the predicted value P_TARGET_FLAG times the average target amount in the training set for the type of car. I was attempting a Probability/Severity calculation.

Conclusion

When dealing with categorical variables, there are a number of techniques which may, or may not, prove beneficial in preparing the data. In this situation, I did not find creating dummy variables for the different categories produced any particular improvement in the models I tried. Reducing the dummy-variable data to eliminate multi-collinearity helped somewhat, but that data path was still less productive than the original data after converting to factors and binning the income data.

The final selected model uses features chosen for their importance as determined by fitting a random forest model to the data. The model uses nearly all of the original features, but it is easy to understand and easy to explain. The prediction results from the selected model are only a slight degradation from the best of the models built. The best model has 2 times the features, and the data preparation will be difficult to explain to anyone not familiar with the dummy-variable process. Simpler models can perform well, and the added complexity is not always worth the extra cost.

Bingo-Bonus Write up

Using the glm logit link with Model6, I got the following results:

	Logit	Model6
AIC	7352.801	7352.801
BIC	7647.101	7647.101
Log Likelihood	7268.801	7268.801
KS Statistic	0.4719	0.4719
AUC	0.8156	0.8156

For the result metrics we are using, the 2 methods performed identically. Even the ROC curve looks the same, which is expected given the other values.