M7: Data Assignment

Jason Merten

Data Exploration/Preparation:

The insurance data sets used in this analysis contains 26 columns of data across a training and a test data set. The training data set consists of 8161 rows of data while the test data set consists of 2141 rows of data. After adjusting the variable types, there are 6 columns that contain missing values: **AGE**, **YOJ**, **INCOME**, **HOME_VAL**, **OLDCLAIM**, and **CAR_AGE**. Most of the variables within the data set appear to follow some sort of statistical distribution (normal, beta, etc.) and a few have some outliers, though nothing extreme (less than 1%). All outliers were smoothed to the 99th and 1st quantile values for all numerical variables. While handling missing values, I decided to create 4 new variables in each data set (**HOME_OWNER**, **SQRT_TRAVTIME**, **SQRT_BLUEBOOK**, and **INCOME_bin**) to help with model accuracy. I also renamed the blank **JOB** level name to 'None' to help reduce confusion. Additionally, the levels for **RED_CAR** and **DO_KIDS_DRIVE** needed to be modified to 'Yes' and 'No' to prevent errors while training the decision trees.

Building Models:

Model1 and Model1.reg were created as standard decision trees using an 80/20 train/test split of the training data set. Model1 is the classification model, attempting to fit to the variable TARGET_FLAG. Model1.reg is the regression model, attempting to fit to the variable TARGET_AMT using 10-fold cross-validation and pruning. Both models had decent performance on the test data set.

Model2 and Model2.1 were created using random forests in the caret package with 10-fold repeated cross-validation over three iterations as well as a grid search of variables. The hyperparameter mtry was selected by iterating through multiple values and selecting the best value to reduce the time to train the final models. An interesting note is that the value of min.node.size must be 1 for classification and 5 for regression.

Model3 and Model3.1 were created using by using a boosted decision tree using the gbm function within the caret package with 5-fold repeated cross-validation over three iterations as well as a grid search of variables. All hyperparameters (n.trees, interaction.depth, shrinkage, and n.minobsinnode) were selected by iterating through multiple values and selecting the best value (maximizing accuracy or R^2) to reduce training time.

Selecting a Model:

To analyze the models, I used the metrics from the confusionMatrix and postResample functions and looked to maximize the average of Accuracy, Sensitivity, and Specificity for classification models and R^2 for regression models. Overall, I picked Model3 and Model2.1 as my champion models. Below is a table of the metrics for each model:

Model/Metric	Accuracy	Sensitivity	Specificity	RMSE	R^2	MAE
Model 1	.6973	.7896	.4505			
Model 2	.7782	.9242	.3874			
Model 3 <<<	.7776	.9049	.4369			
Model1.reg				5542.147	.0135	2306.847
Model2.1 <<<				5424.279	.0585	2170.102
Model3.1				5449.978	.0454	2138.192