BHao_HW4

DATA EXPLORATION

- Summary data indicate that 26.38% of records reported car crashes, which on average resulted in $5,702.18 in damages

- Given that the dollar figures were imported as strings, we must first eliminate the dollar signs and commas and then convert to numbers in order to do much data exploration

- Also, there are several variables that have NAs, i.e. AGE, YOJ, INCOME, HOME_VAL, CAR_AGE; we'll likely use caret's built-in preprocessing functions to address these NAs

- We'll drop the INDEX variable here

```r
insur = read.csv('insurance_training_data.csv', stringsAsFactors = TRUE)
str(insur)
```

```
## 'data.frame':    8161 obs. of  26 variables:
##  $ INDEX      : int  1 2 4 5 6 7 8 11 12 13 ...
##  $ TARGET_FLAG: int  0 0 0 0 0 1 0 1 1 0 ...
##  $ TARGET_AMT : num  0 0 0 0 0 ...
##  $ KIDSDRIV   : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ AGE        : int  60 43 35 51 50 34 54 37 34 50 ...
##  $ HOMEKIDS   : int  0 0 1 0 0 1 0 2 0 0 ...
##  $ YOJ        : int  11 11 10 14 NA 12 NA NA 10 7 ...
##  $ INCOME     : Factor w/ 6613 levels "","$0","$1,007",..: 5033 6292 1250 1 509 746 1488 315 4765 28
##  $ PARENT1    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 1 1 ...
##  $ HOME_VAL   : Factor w/ 5107 levels "","$0","$100,093",..: 2 3259 348 3917 3034 2 1 4167 2 2 ...
##  $ MSTATUS    : Factor w/ 2 levels "Yes","z_No": 2 2 1 1 1 2 1 1 2 2 ...
##  $ SEX        : Factor w/ 2 levels "M","z_F": 1 1 2 1 2 2 2 1 2 1 ...
##  $ EDUCATION  : Factor w/ 5 levels "<High School",..: 4 5 5 1 4 2 1 2 2 2 ...
##  $ JOB        : Factor w/ 9 levels "","Clerical",..: 7 9 2 9 3 9 9 9 2 7 ...
##  $ TRAVTIME   : int  14 22 5 32 36 46 33 44 34 48 ...
##  $ CAR_USE    : Factor w/ 2 levels "Commercial","Private": 2 1 2 2 2 1 2 1 2 1 ...
##  $ BLUEBOOK   : Factor w/ 2789 levels "$1,500","$1,520",..: 434 503 2212 553 802 746 2672 701 135 85
##  $ TIF        : int  11 1 4 7 1 1 1 1 1 7 ...
##  $ CAR_TYPE   : Factor w/ 6 levels "Minivan","Panel Truck",..: 1 1 6 1 6 4 6 5 6 5 ...
##  $ RED_CAR    : Factor w/ 2 levels "no","yes": 2 2 1 2 1 1 1 2 1 1 ...
##  $ OLDCLAIM   : Factor w/ 2857 levels "$0","$1,000",..: 1449 1 1311 1 432 1 1 510 1 1 ...
##  $ CLM_FREQ   : int  2 0 2 0 2 0 0 1 0 0 ...
##  $ REVOKED    : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 1 2 1 1 ...
##  $ MVR_PTS    : int  3 0 3 0 3 0 0 10 0 1 ...
##  $ CAR_AGE    : int  18 1 10 6 17 7 1 7 1 17 ...
##  $ URBANICITY : Factor w/ 2 levels "Highly Urban/ Urban",..: 1 1 1 1 1 1 1 1 1 2 ...
```

```r
# drop index column
insur = subset(insur, select = -c(INDEX))

# eliminate dollar signs and commas and convert to numbers
insur$INCOME = as.numeric(sub('\\$', '', sub('\\,', '', insur$INCOME)))
insur$HOME_VAL = as.numeric(sub('\\$', '', sub('\\,', '', insur$HOME_VAL)))
insur$BLUEBOOK = as.numeric(sub('\\$', '', sub('\\,', '', insur$BLUEBOOK)))
```

```r
insur$OLDCLAIM = as.numeric(sub('\\$', '', sub('\\,', '', insur$OLDCLAIM)))

summary(insur)
```

```
##   TARGET_FLAG       TARGET_AMT        KIDSDRIV           AGE
##  Min.   :0.0000   Min.   :     0   Min.   :0.0000   Min.   :16.00
##  1st Qu.:0.0000   1st Qu.:     0   1st Qu.:0.0000   1st Qu.:39.00
##  Median :0.0000   Median :     0   Median :0.0000   Median :45.00
##  Mean   :0.2638   Mean   :  1504   Mean   :0.1711   Mean   :44.79
##  3rd Qu.:1.0000   3rd Qu.:  1036   3rd Qu.:0.0000   3rd Qu.:51.00
##  Max.   :1.0000   Max.   :107586   Max.   :4.0000   Max.   :81.00
##                                                     NA's   :6
##     HOMEKIDS           YOJ           INCOME        PARENT1
##  Min.   :0.0000   Min.   : 0.0   Min.   :     0   No :7084
##  1st Qu.:0.0000   1st Qu.: 9.0   1st Qu.: 28097   Yes:1077
##  Median :0.0000   Median :11.0   Median : 54028
##  Mean   :0.7212   Mean   :10.5   Mean   : 61898
##  3rd Qu.:1.0000   3rd Qu.:13.0   3rd Qu.: 85986
##  Max.   :5.0000   Max.   :23.0   Max.   :367030
##                   NA's   :454    NA's   :445
##     HOME_VAL      MSTATUS      SEX               EDUCATION
##  Min.   :     0   Yes :4894   M  :3786   <High School :1203
##  1st Qu.:     0   z_No:3267   z_F:4375   Bachelors    :2242
##  Median :161160                          Masters      :1658
##  Mean   :154867                          PhD          : 728
##  3rd Qu.:238724                          z_High School:2330
##  Max.   :885282
##  NA's   :464
##             JOB          TRAVTIME            CAR_USE        BLUEBOOK
##  z_Blue Collar:1825   Min.   :  5.00   Commercial:3029   Min.   : 1500
##  Clerical     :1271   1st Qu.: 22.00   Private   :5132   1st Qu.: 9280
##  Professional :1117   Median : 33.00                     Median :14440
##  Manager      : 988   Mean   : 33.49                     Mean   :15710
##  Lawyer       : 835   3rd Qu.: 44.00                     3rd Qu.:20850
##  Student      : 712   Max.   :142.00                     Max.   :69740
##  (Other)      :1413
##       TIF                CAR_TYPE    RED_CAR       OLDCLAIM
##  Min.   : 1.000   Minivan    :2145   no :5783   Min.   :    0
##  1st Qu.: 1.000   Panel Truck: 676   yes:2378   1st Qu.:    0
##  Median : 4.000   Pickup     :1389              Median :    0
##  Mean   : 5.351   Sports Car : 907              Mean   : 4037
##  3rd Qu.: 7.000   Van        : 750              3rd Qu.: 4636
##  Max.   :25.000   z_SUV      :2294              Max.   :57037
##
##     CLM_FREQ      REVOKED       MVR_PTS          CAR_AGE
##  Min.   :0.0000   No :7161   Min.   : 0.000   Min.   :-3.000
##  1st Qu.:0.0000   Yes:1000   1st Qu.: 0.000   1st Qu.: 1.000
##  Median :0.0000              Median : 1.000   Median : 8.000
##  Mean   :0.7986              Mean   : 1.696   Mean   : 8.328
##  3rd Qu.:2.0000              3rd Qu.: 3.000   3rd Qu.:12.000
##  Max.   :5.0000              Max.   :13.000   Max.   :28.000
##                                               NA's   :510
##                  URBANICITY
##  Highly Urban/ Urban  :6492
```

```
##  z_Highly Rural/ Rural:1669
##
##
##
##
##
```

```r
# mean of actual accidents
mean(insur[insur$TARGET_FLAG == 1, 'TARGET_AMT'])
```

```
## [1] 5702.18
```

DATA PREPARATION

- Since I'll be using the caret package for modeling, I need to convert the categorical variables to factors
- We'll also use caret's medianImpute method within the preProcessing function to handle NAs
- Well, the medianImpute preprocessing step isn't working as expected, so I'll manually impute the medians for NA data
- Lastly, we'll use caret's center and scale preprocessing features to automatically transform the data during modeling

```r
# convert categorical variables to factors
insur$TARGET_FLAG = factor(insur$TARGET_FLAG, labels = c('no_crash', 'yes_crash'))
str(insur)
```

```
## 'data.frame':    8161 obs. of  25 variables:
##  $ TARGET_FLAG: Factor w/ 2 levels "no_crash","yes_crash": 1 1 1 1 1 2 1 2 2 1 ...
##  $ TARGET_AMT : num  0 0 0 0 0 ...
##  $ KIDSDRIV   : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ AGE        : int  60 43 35 51 50 34 54 37 34 50 ...
##  $ HOMEKIDS   : int  0 0 1 0 0 1 0 2 0 0 ...
##  $ YOJ        : int  11 11 10 14 NA 12 NA NA 10 7 ...
##  $ INCOME     : num  67349 91449 16039 NA 114986 ...
##  $ PARENT1    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 2 1 1 1 1 ...
##  $ HOME_VAL   : num  0 257252 124191 306251 243925 ...
##  $ MSTATUS    : Factor w/ 2 levels "Yes","z_No": 2 2 1 1 1 2 1 1 2 2 ...
##  $ SEX        : Factor w/ 2 levels "M","z_F": 1 1 2 1 2 2 2 1 2 1 ...
##  $ EDUCATION  : Factor w/ 5 levels "<High School",..: 4 5 5 1 4 2 1 2 2 2 ...
##  $ JOB        : Factor w/ 9 levels "","Clerical",..: 7 9 2 9 3 9 9 9 2 7 ...
##  $ TRAVTIME   : int  14 22 5 32 36 46 33 44 34 48 ...
##  $ CAR_USE    : Factor w/ 2 levels "Commercial","Private": 2 1 2 2 2 1 2 1 2 1 ...
##  $ BLUEBOOK   : num  14230 14940 4010 15440 18000 ...
##  $ TIF        : int  11 1 4 7 1 1 1 1 1 7 ...
##  $ CAR_TYPE   : Factor w/ 6 levels "Minivan","Panel Truck",..: 1 1 6 1 6 4 6 5 6 5 ...
##  $ RED_CAR    : Factor w/ 2 levels "no","yes": 2 2 1 2 1 1 1 2 1 1 ...
##  $ OLDCLAIM   : num  4461 0 38690 0 19217 ...
##  $ CLM_FREQ   : int  2 0 2 0 2 0 0 1 0 0 ...
##  $ REVOKED    : Factor w/ 2 levels "No","Yes": 1 1 1 1 2 1 1 2 1 1 ...
##  $ MVR_PTS    : int  3 0 3 0 3 0 0 10 0 1 ...
##  $ CAR_AGE    : int  18 1 10 6 17 7 1 7 1 17 ...
##  $ URBANICITY : Factor w/ 2 levels "Highly Urban/ Urban",..: 1 1 1 1 1 1 1 1 1 2 ...
```

```r
# median imputation
impute_median = function(x) replace(x, is.na(x), median(x, na.rm = TRUE))
insur$AGE = impute_median(insur$AGE)
insur$YOJ = impute_median(insur$YOJ)
insur$INCOME = impute_median(insur$INCOME)
insur$HOME_VAL = impute_median(insur$HOME_VAL)
insur$CAR_AGE = impute_median(insur$CAR_AGE)
```

BUILD MODELS

- We'll first build a linear regression model to estimate the cost of damage for those records with accidents
- Then, we'll build a logistic regression model to predict the likelihood of an accident for all records

BUILD MODELS - LINEAR REGRESSION

- For the linear regression, we'll just use those records where an accident occurred
- In terms of setup, we are using 10-fold cross validation to measure out-of-sample performance and are using the same folds for each model to ensure comparable results
- We then start by including all variables and then remove statistically insignificant ones at the 5% level until all remaining are significant
- We then tried a glmnet model which combines lasso and ridge regression; given that it penalizes large magnitude and the number of non-zero coefficients, it can be used for variable selection
- Lastly, we fit a random forest model just for fun
- Based on the RMSE dot plot, the simple linear regression model based on only on BLUEBOOK performed the best and is our final model
- You can also see the improvement as variables were removed; also note how well the glmnet rf model performed without manual tuning

```r
library(caret)
library(caretEnsemble)

# drop TARGET_AMT variable
insur_acc = insur[insur$TARGET_FLAG == 'yes_crash',]
insur_acc = subset(insur_acc, select = -c(TARGET_FLAG))

set.seed(123)
# use cross validation to compare out-of-sample ROC for all models
# use the same folds for each model to ensure comparable results
myFolds = createFolds(insur_acc$TARGET_AMT, k = 10)

# used instead of method = 'cv', number = 10
myControl = trainControl(verboseIter = FALSE,savePredictions = TRUE, index = myFolds)

# model using glm model
model_glm_full = train(TARGET_AMT ~ ., data = insur_acc, metric = 'RMSE', method = 'glm',
```

```r
                  preProcess = c('center', 'scale'), trControl = myControl)
summary(model_glm_full)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -8949   -3175   -1502     481   99588
##
## Coefficients:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          5702.180    165.741  34.404  < 2e-16 ***
## KIDSDRIV                             -107.299    198.487  -0.541   0.5889
## AGE                                   174.950    202.942   0.862   0.3887
## HOMEKIDS                              254.837    247.630   1.029   0.3035
## YOJ                                    84.096    215.761   0.390   0.6968
## INCOME                               -375.155    281.031  -1.335   0.1820
## PARENT1Yes                            115.748    243.768   0.475   0.6350
## HOME_VAL                              252.216    232.888   1.083   0.2789
## MSTATUSz_No                           400.354    246.746   1.623   0.1048
## SEXz_F                               -697.421    326.381  -2.137   0.0327 *
## EDUCATIONBachelors                    106.952    275.122   0.389   0.6975
## EDUCATIONMasters                      422.069    388.882   1.085   0.2779
## EDUCATIONPhD                          556.095    306.746   1.813   0.0700 .
## `EDUCATIONz_High School`             -192.022    248.239  -0.774   0.4393
## JOBClerical                           117.170    454.416   0.258   0.7965
## JOBDoctor                            -243.678    203.190  -1.199   0.2306
## `JOBHome Maker`                        -5.943    350.608  -0.017   0.9865
## JOBLawyer                              84.218    264.531   0.318   0.7502
## JOBManager                           -189.951    260.251  -0.730   0.4655
## JOBProfessional                       337.470    359.916   0.938   0.3485
## JOBStudent                             37.535    423.233   0.089   0.9293
## `JOBz_Blue Collar`                    237.155    522.508   0.454   0.6500
## TRAVTIME                               10.948    168.366   0.065   0.9482
## CAR_USEPrivate                       -220.715    260.776  -0.846   0.3974
## BLUEBOOK                             1033.610    253.431   4.078  4.7e-05 ***
## TIF                                   -61.793    167.218  -0.370   0.7118
## `CAR_TYPEPanel Truck`                -177.220    264.580  -0.670   0.5030
## CAR_TYPEPickup                        -24.541    241.323  -0.102   0.9190
## `CAR_TYPESports Car`                  370.826    261.310   1.419   0.1560
## CAR_TYPEVan                            18.066    224.295   0.081   0.9358
## CAR_TYPEz_SUV                         420.744    309.779   1.358   0.1745
## RED_CARyes                            -87.254    224.430  -0.389   0.6975
## OLDCLAIM                              250.837    227.964   1.100   0.2713
## CLM_FREQ                             -144.850    197.269  -0.734   0.4629
## REVOKEDYes                           -455.247    208.909  -2.179   0.0294 *
## MVR_PTS                               286.741    176.742   1.622   0.1049
## CAR_AGE                              -516.969    235.250  -2.198   0.0281 *
## `URBANICITYz_Highly Rural/ Rural`     -21.816    170.088  -0.128   0.8980
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for gaussian family taken to be 59143116)
##
##     Null deviance: 1.2903e+11  on 2152  degrees of freedom
## Residual deviance: 1.2509e+11  on 2115  degrees of freedom
## AIC: 44679
##
## Number of Fisher Scoring iterations: 2
```

```r
# let's drop any statistically insignificant variables at 5%
model_glm_sig1 = train(TARGET_AMT ~ SEX + BLUEBOOK + REVOKED + CAR_AGE, data = insur_acc,
                       metric = 'RMSE', method = 'glm', preProcess = c('center', 'scale'),
                       trControl = myControl)
summary(model_glm_sig1)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -7775   -3134   -1578     390  100747
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5702.2      165.5  34.451  < 2e-16 ***
## SEXz_F        -312.1      166.3  -1.876   0.0607 .
## BLUEBOOK       933.5      169.1   5.522 3.76e-08 ***
## REVOKEDYes    -277.3      165.6  -1.675   0.0941 .
## CAR_AGE       -260.4      168.5  -1.545   0.1224
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 58982559)
##
##     Null deviance: 1.2903e+11  on 2152  degrees of freedom
## Residual deviance: 1.2669e+11  on 2148  degrees of freedom
## AIC: 44640
##
## Number of Fisher Scoring iterations: 2
```

```r
# let's again drop any additional statistically insigificant variables at 5%
model_glm_sig2 = train(TARGET_AMT ~ BLUEBOOK, data = insur_acc,
                       metric = 'RMSE', method = 'glm', preProcess = c('center', 'scale'),
                       trControl = myControl)
summary(model_glm_sig2)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -7757   -3083   -1541     295  101459
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5702.2      165.7  34.403  < 2e-16 ***
## BLUEBOOK       914.4      165.8   5.515  3.9e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 59148308)
##
##     Null deviance: 1.2903e+11  on 2152  degrees of freedom
## Residual deviance: 1.2723e+11  on 2151  degrees of freedom
## AIC: 44643
##
## Number of Fisher Scoring iterations: 2
```

```
# let's try a glmnet model that combines ridge vs. lasso regression
# since it penalizes either or both magnitude and number of non-zero coefficients, it can be used for v
model_glmnet = train(TARGET_AMT ~ ., data = insur_acc, metric = 'RMSE', method = 'glmnet',
                     preProcess = c('center', 'scale'), trControl = myControl)
coef(model_glmnet$finalModel, s = model_glmnet$finalModel$tuneValue$lambda)
```

```
## 38 x 1 sparse Matrix of class "dgCMatrix"
##                                       1
## (Intercept)                 5702.179960
## KIDSDRIV                       .
## AGE                            .
## HOMEKIDS                       .
## YOJ                            .
## INCOME                         .
## PARENT1Yes                     4.680203
## HOME_VAL                       .
## MSTATUSz_No                   72.531158
## SEXz_F                      -153.865326
## EDUCATIONBachelors             .
## EDUCATIONMasters               .
## EDUCATIONPhD                   .
## EDUCATIONz_High School        -3.731557
## JOBClerical                    .
## JOBDoctor                      .
## JOBHome Maker                  .
## JOBLawyer                      .
## JOBManager                   -60.776693
## JOBProfessional               27.963950
## JOBStudent                     .
## JOBz_Blue Collar               .
## TRAVTIME                       .
## CAR_USEPrivate                 .
## BLUEBOOK                     727.639154
## TIF                            .
## CAR_TYPEPanel Truck            .
## CAR_TYPEPickup                 .
## CAR_TYPESports Car             .
## CAR_TYPEVan                    .
## CAR_TYPEz_SUV                  .
## RED_CARyes                     .
## OLDCLAIM                       .
```

```
## CLM_FREQ                               .
## REVOKEDYes                      -100.899876
## MVR_PTS                          138.696348
## CAR_AGE                          -36.564038
## URBANICITYz_Highly Rural/ Rural    .
```

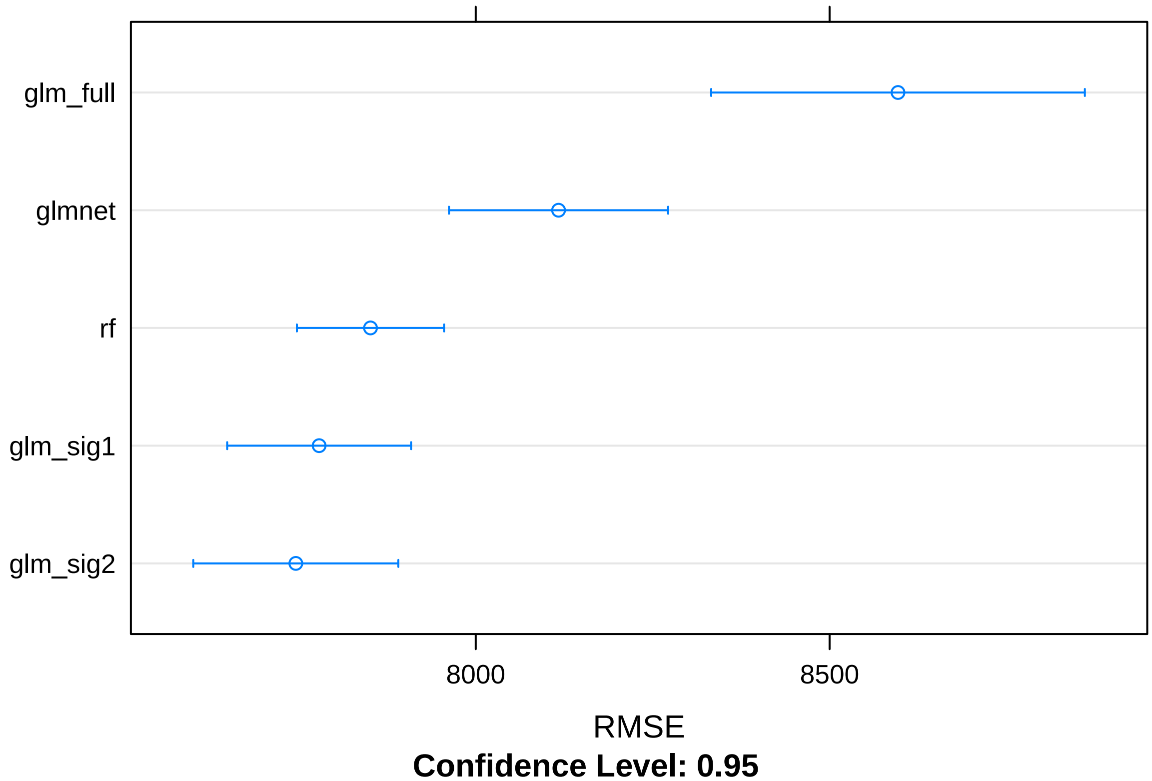
```r
# let's also model using random forest just for fun
model_rf = train(TARGET_AMT ~ ., data = insur_acc, metric = 'RMSE', method = 'ranger', trControl = myCon

# compare models
model_list = list(glm_full = model_glm_full, glm_sig1 = model_glm_sig1, glm_sig2 = model_glm_sig2,
                  glmnet = model_glmnet, rf = model_rf)

# collect resamples from the CV folds
resamps = resamples(model_list)
summary(resamps)
```

```
##
## Call:
## summary.resamples(object = resamps)
##
## Models: glm_full, glm_sig1, glm_sig2, glmnet, rf
## Number of resamples: 10
##
## RMSE
##          Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
## glm_full 8254    8298   8459 8597    8848 9213    0
## glm_sig1 7459    7621   7843 7779    7894 8000    0
## glm_sig2 7431    7567   7801 7746    7894 7998    0
## glmnet   7867    7984   8056 8117    8215 8519    0
## rf       7587    7757   7903 7851    7940 8012    0
##
## Rsquared
##               Min.   1st Qu.    Median     Mean  3rd Qu.     Max. NA's
## glm_full 1.036e-05 0.0003173 0.0004688 0.001226 0.001097 0.004556    0
## glm_sig1 5.612e-03 0.0063330 0.0084050 0.008895 0.010830 0.014110    0
## glm_sig2 8.973e-03 0.0128700 0.0139600 0.014050 0.015610 0.017250    0
## glmnet   4.329e-05 0.0004266 0.0015540 0.001634 0.002342 0.003964    0
## rf       1.789e-05 0.0002911 0.0006356 0.001009 0.001147 0.003580    0
```

```r
dotplot(resamps, metric = 'RMSE')
```



BUILD MODELS - LOGISTIC REGRESSION

- For the logistic regression model, we'll drop the TARGET_AMT
- In terms of setup, we are using 10-fold cross validation to measure out-of-sample performance and are using the same folds for each model to ensure comparable results
- We then start by including all variables and then remove statistically insignificant ones at the 5% level until all remaining are significant
- We then tried a glmnet model which combines lasso and ridge regression; given that it penalizes large magnitude and the number of non-zero coefficients, it can be used for variable selection
- Lastly, we fit a random forest model just for fun
- Based on the ROC dot plot (the x-axis is actually AUC), the logistic regression model based on two backwardation steps performed the best and is our final selected model
- You can also see the improvement as variables were removed; also note how well the glmnet rf model performed without manual tuning

```
# drop TARGET_AMT variable
insur = subset(insur, select = -c(TARGET_AMT))

set.seed(123)
# use cross validation to compare out-of-sample ROC for all models
```

```r
# use the same folds for each model to ensure comparable results
myFolds = createFolds(insur$TARGET_FLAG, k = 10)

# confirm that folks preserve class distribution
table(insur$TARGET_FLAG) / nrow(insur)
```

```
##
##  no_crash yes_crash
## 0.7361843 0.2638157
```

```r
table(insur$TARGET_FLAG[myFolds$Fold1]) / length(myFolds$Fold1)
```

```
##
##  no_crash yes_crash
## 0.7365196 0.2634804
```

```r
myControl = trainControl(summaryFunction = twoClassSummary, classProbs = TRUE, verboseIter = FALSE,
                         savePredictions = TRUE, index = myFolds)  # used instead of method = 'cv', num

# model using glm model
model_glm_full = train(TARGET_FLAG ~ ., data = insur, metric = 'ROC', method = 'glm',
                  preProcess = c('center', 'scale'), trControl = myControl,
                  na.action = na.pass)
summary(model_glm_full)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5850  -0.7127  -0.3983   0.6264   3.1526
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -1.429906   0.034926 -40.941  < 2e-16
## KIDSDRIV                         0.197555   0.031317   6.308 2.82e-10
## AGE                             -0.008734   0.034669  -0.252 0.801103
## HOMEKIDS                         0.055422   0.041448   1.337 0.181171
## YOJ                             -0.044001   0.034145  -1.289 0.197521
## INCOME                          -0.158363   0.050067  -3.163 0.001562
## PARENT1Yes                       0.129305   0.037100   3.485 0.000492
## HOME_VAL                        -0.163878   0.042888  -3.821 0.000133
## MSTATUSz_No                      0.241944   0.040951   5.908 3.46e-09
## SEXz_F                          -0.041120   0.055872  -0.736 0.461749
## EDUCATIONBachelors              -0.169361   0.051618  -3.281 0.001034
## EDUCATIONMasters                -0.115355   0.071923  -1.604 0.108742
## EDUCATIONPhD                    -0.046785   0.060979  -0.767 0.442943
## `EDUCATIONz_High School`         0.008126   0.042934   0.189 0.849879
## JOBClerical                      0.148971   0.071313   2.089 0.036709
## JOBDoctor                       -0.076212   0.045670  -1.669 0.095163
## `JOBHome Maker`                  0.062556   0.056542   1.106 0.268573
## JOBLawyer                        0.031811   0.051364   0.619 0.535705
## JOBManager                      -0.181762   0.055963  -3.248 0.001163
## JOBProfessional                  0.055667   0.061331   0.908 0.364065
```

```
## JOBStudent                          0.061030   0.060532   1.008 0.313345
## `JOBz_Blue Collar`                  0.129468   0.077319   1.674 0.094041
## TRAVTIME                            0.231754   0.029958   7.736 1.03e-14
## CAR_USEPrivate                     -0.365450   0.044313  -8.247  < 2e-16
## BLUEBOOK                           -0.175470   0.044313  -3.960 7.50e-05
## TIF                                -0.229985   0.030451  -7.553 4.27e-14
## `CAR_TYPEPanel Truck`               0.154558   0.044595   3.466 0.000529
## CAR_TYPEPickup                      0.208196   0.037856   5.500 3.81e-08
## `CAR_TYPESports Car`                0.322190   0.040824   7.892 2.97e-15
## CAR_TYPEVan                         0.178685   0.036538   4.890 1.01e-06
## CAR_TYPEz_SUV                       0.345333   0.050020   6.904 5.06e-12
## RED_CARyes                         -0.004404   0.039244  -0.112 0.910644
## OLDCLAIM                           -0.121951   0.034318  -3.554 0.000380
## CLM_FREQ                            0.227005   0.033068   6.865 6.66e-12
## REVOKEDYes                          0.290990   0.029950   9.716  < 2e-16
## MVR_PTS                             0.243258   0.029225   8.324  < 2e-16
## CAR_AGE                            -0.005965   0.041627  -0.143 0.886064
## `URBANICITYz_Highly Rural/ Rural`  -0.963973   0.045511 -21.181  < 2e-16
##
## (Intercept)                        ***
## KIDSDRIV                           ***
## AGE
## HOMEKIDS
## YOJ
## INCOME                             **
## PARENT1Yes                         ***
## HOME_VAL                           ***
## MSTATUSz_No                        ***
## SEXz_F
## EDUCATIONBachelors                 **
## EDUCATIONMasters
## EDUCATIONPhD
## `EDUCATIONz_High School`
## JOBClerical                        *
## JOBDoctor                          .
## `JOBHome Maker`
## JOBLawyer
## JOBManager                         **
## JOBProfessional
## JOBStudent
## `JOBz_Blue Collar`                 .
## TRAVTIME                           ***
## CAR_USEPrivate                     ***
## BLUEBOOK                           ***
## TIF                                ***
## `CAR_TYPEPanel Truck`              ***
## CAR_TYPEPickup                     ***
## `CAR_TYPESports Car`               ***
## CAR_TYPEVan                        ***
## CAR_TYPEz_SUV                      ***
## RED_CARyes
## OLDCLAIM                           ***
## CLM_FREQ                           ***
## REVOKEDYes                         ***
```

```
## MVR_PTS                          ***
## CAR_AGE
## `URBANICITYz_Highly Rural/ Rural` ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9418.0  on 8160  degrees of freedom
## Residual deviance: 7297.6  on 8123  degrees of freedom
## AIC: 7373.6
##
## Number of Fisher Scoring iterations: 5
```

```
# let's drop any statistically insignificant variables at 5%
model_glm_sig1 = train(TARGET_FLAG ~ KIDSDRIV + INCOME + PARENT1 + HOME_VAL + MSTATUS +
                       EDUCATION + JOB + TRAVTIME + CAR_USE + BLUEBOOK + TIF + CAR_TYPE +
                       OLDCLAIM + CLM_FREQ + REVOKED + MVR_PTS + URBANICITY, data = insur,
                       metric = 'ROC', method = 'glm', preProcess = c('center', 'scale'),
                       trControl = myControl, na.action = na.pass)
summary(model_glm_sig1)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6039  -0.7115  -0.3979   0.6268   3.1440
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -1.429203   0.034913 -40.937  < 2e-16
## KIDSDRIV                          0.213612   0.028197   7.576 3.57e-14
## INCOME                           -0.161385   0.049824  -3.239 0.001199
## PARENT1Yes                        0.155761   0.031908   4.882 1.05e-06
## HOME_VAL                         -0.168299   0.042725  -3.939 8.18e-05
## MSTATUSz_No                       0.231250   0.038981   5.932 2.99e-09
## EDUCATIONBachelors               -0.172675   0.048592  -3.554 0.000380
## EDUCATIONMasters                 -0.122020   0.064974  -1.878 0.060385
## EDUCATIONPhD                     -0.051812   0.057055  -0.908 0.363825
## `EDUCATIONz_High School`          0.006716   0.042772   0.157 0.875229
## JOBClerical                       0.150170   0.071267   2.107 0.035104
## JOBDoctor                        -0.076524   0.045604  -1.678 0.093344
## `JOBHome Maker`                   0.073942   0.054926   1.346 0.178234
## JOBLawyer                         0.029446   0.051269   0.574 0.565740
## JOBManager                       -0.184288   0.055911  -3.296 0.000980
## JOBProfessional                   0.053224   0.061306   0.868 0.385304
## JOBStudent                        0.077634   0.059513   1.304 0.192066
## `JOBz_Blue Collar`                0.129101   0.077296   1.670 0.094879
## TRAVTIME                          0.230338   0.029919   7.699 1.37e-14
## CAR_USEPrivate                   -0.365946   0.044259  -8.268  < 2e-16
## BLUEBOOK                         -0.194343   0.039735  -4.891 1.00e-06
## TIF                              -0.229633   0.030435  -7.545 4.53e-14
## `CAR_TYPEPanel Truck`             0.167880   0.041606   4.035 5.46e-05
```

```
## CAR_TYPEPickup                    0.206811   0.037815   5.469 4.53e-08
## `CAR_TYPESports Car`              0.305697   0.033764   9.054  < 2e-16
## CAR_TYPEVan                       0.186801   0.035282   5.295 1.19e-07
## CAR_TYPEz_SUV                     0.321689   0.038645   8.324  < 2e-16
## OLDCLAIM                         -0.123293   0.034295  -3.595 0.000324
## CLM_FREQ                          0.227369   0.033040   6.882 5.91e-12
## REVOKEDYes                        0.292737   0.029917   9.785  < 2e-16
## MVR_PTS                           0.245383   0.029172   8.412  < 2e-16
## `URBANICITYz_Highly Rural/ Rural` -0.963765   0.045501 -21.181  < 2e-16
##
## (Intercept)                      ***
## KIDSDRIV                         ***
## INCOME                           **
## PARENT1Yes                       ***
## HOME_VAL                         ***
## MSTATUSz_No                      ***
## EDUCATIONBachelors               ***
## EDUCATIONMasters                 .
## EDUCATIONPhD
## `EDUCATIONz_High School`
## JOBClerical                      *
## JOBDoctor                        .
## `JOBHome Maker`
## JOBLawyer
## JOBManager                       ***
## JOBProfessional
## JOBStudent
## `JOBz_Blue Collar`               .
## TRAVTIME                         ***
## CAR_USEPrivate                   ***
## BLUEBOOK                         ***
## TIF                              ***
## `CAR_TYPEPanel Truck`            ***
## CAR_TYPEPickup                   ***
## `CAR_TYPESports Car`             ***
## CAR_TYPEVan                      ***
## CAR_TYPEz_SUV                    ***
## OLDCLAIM                         ***
## CLM_FREQ                         ***
## REVOKEDYes                       ***
## MVR_PTS                          ***
## `URBANICITYz_Highly Rural/ Rural` ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9418.0  on 8160  degrees of freedom
## Residual deviance: 7301.8  on 8129  degrees of freedom
## AIC: 7365.8
##
## Number of Fisher Scoring iterations: 5
```

```r
# let's again drop any additional statistically insigificant variables at 5%
model_glm_sig2 = train(TARGET_FLAG ~ KIDSDRIV + INCOME + PARENT1 + HOME_VAL + MSTATUS +
                       TRAVTIME + CAR_USE + BLUEBOOK + TIF + CAR_TYPE +
                       OLDCLAIM + CLM_FREQ + REVOKED + MVR_PTS + URBANICITY, data = insur,
                       metric = 'ROC', method = 'glm', preProcess = c('center', 'scale'),
                       trControl = myControl, na.action = na.pass)
summary(model_glm_sig2)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4802  -0.7381  -0.4165   0.6566   3.0520
##
## Coefficients:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)                         -1.40883    0.03449 -40.851  < 2e-16 ***
## KIDSDRIV                             0.21428    0.02790   7.680 1.59e-14 ***
## INCOME                              -0.34018    0.04067  -8.365  < 2e-16 ***
## PARENT1Yes                           0.16523    0.03140   5.261 1.43e-07 ***
## HOME_VAL                            -0.18563    0.04138  -4.486 7.25e-06 ***
## MSTATUSz_No                          0.20075    0.03821   5.253 1.49e-07 ***
## TRAVTIME                             0.23495    0.02952   7.958 1.75e-15 ***
## CAR_USEPrivate                      -0.43453    0.03343 -12.996  < 2e-16 ***
## BLUEBOOK                            -0.21301    0.03939  -5.408 6.39e-08 ***
## TIF                                 -0.22419    0.03010  -7.449 9.41e-14 ***
## `CAR_TYPEPanel Truck`                0.13557    0.03861   3.511 0.000446 ***
## CAR_TYPEPickup                       0.18284    0.03644   5.017 5.24e-07 ***
## `CAR_TYPESports Car`                 0.29235    0.03304   8.849  < 2e-16 ***
## CAR_TYPEVan                          0.16559    0.03427   4.832 1.35e-06 ***
## CAR_TYPEz_SUV                        0.31487    0.03788   8.312  < 2e-16 ***
## OLDCLAIM                            -0.12042    0.03383  -3.560 0.000371 ***
## CLM_FREQ                             0.22060    0.03262   6.763 1.35e-11 ***
## REVOKEDYes                           0.29306    0.02951   9.932  < 2e-16 ***
## MVR_PTS                              0.25290    0.02887   8.760  < 2e-16 ***
## `URBANICITYz_Highly Rural/ Rural` -0.91146    0.04520 -20.166  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9418  on 8160  degrees of freedom
## Residual deviance: 7430  on 8141  degrees of freedom
## AIC: 7470
##
## Number of Fisher Scoring iterations: 5
```

```r
# let's try a glmnet model that combines ridge vs. lasso regression
# since it penalizes either or both magnitude and number of non-zero coefficients, it can be used for v
model_glmnet = train(TARGET_FLAG ~ ., data = insur, metric = 'ROC', method = 'glmnet',
                     preProcess = c('center', 'scale'), trControl = myControl,
                     na.action = na.pass)
```

```r
coef(model_glmnet$finalModel, s = model_glmnet$finalModel$tuneValue$lambda)
```

```
## 38 x 1 sparse Matrix of class "dgCMatrix"
##                                       1
## (Intercept)                -1.298016977
## KIDSDRIV                    0.153997067
## AGE                        -0.010977711
## HOMEKIDS                    0.054576381
## YOJ                        -0.041316055
## INCOME                     -0.147639490
## PARENT1Yes                  0.119352108
## HOME_VAL                   -0.162670319
## MSTATUSz_No                 0.183002213
## SEXz_F                      .
## EDUCATIONBachelors         -0.087735861
## EDUCATIONMasters           -0.052828368
## EDUCATIONPhD               -0.015111550
## EDUCATIONz_High School      0.045435468
## JOBClerical                 0.061962022
## JOBDoctor                  -0.068806829
## JOBHome Maker               0.002536202
## JOBLawyer                  -0.012118641
## JOBManager                 -0.183430333
## JOBProfessional             .
## JOBStudent                  0.008202489
## JOBz_Blue Collar            0.066996073
## TRAVTIME                    0.167979618
## CAR_USEPrivate             -0.294089312
## BLUEBOOK                   -0.141769905
## TIF                        -0.180239635
## CAR_TYPEPanel Truck         0.071480838
## CAR_TYPEPickup              0.111679610
## CAR_TYPESports Car          0.191047510
## CAR_TYPEVan                 0.088229854
## CAR_TYPEz_SUV               0.188085875
## RED_CARyes                  .
## OLDCLAIM                   -0.046683412
## CLM_FREQ                    0.195173071
## REVOKEDYes                  0.228138587
## MVR_PTS                     0.222881146
## CAR_AGE                    -0.033368342
## URBANICITYz_Highly Rural/ Rural -0.709747658
```

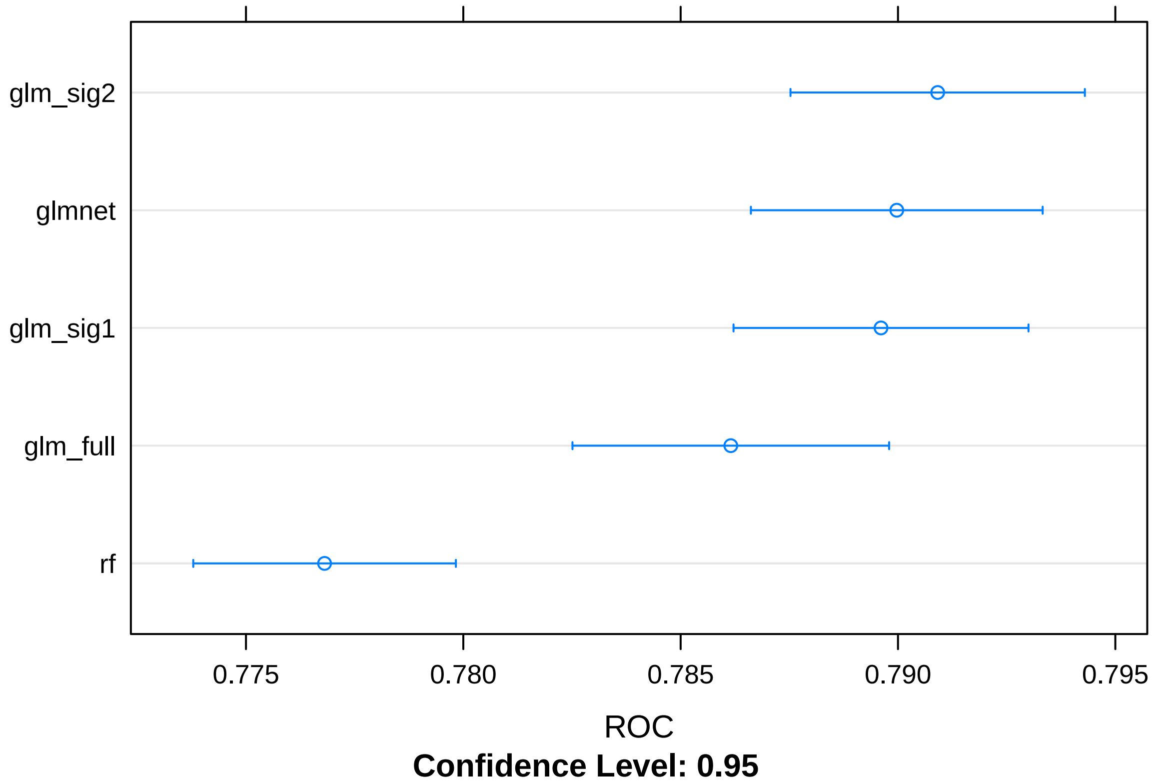
```r
# let's also model using random forest just for fun
model_rf = train(TARGET_FLAG ~ ., data = insur, metric = 'ROC', method = 'ranger',
                 preProcess = c('medianImpute'), trControl = myControl, na.action = na.pass)

# compare models
model_list = list(glm_full = model_glm_full, glm_sig1 = model_glm_sig1, glm_sig2 = model_glm_sig2,
                  glmnet = model_glmnet, rf = model_rf)

# collect resamples from the CV folds
resamps = resamples(model_list)
summary(resamps)
```

```
##
## Call:
## summary.resamples(object = resamps)
##
## Models: glm_full, glm_sig1, glm_sig2, glmnet, rf
## Number of resamples: 10
##
## ROC
##              Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
## glm_full 0.7777  0.7830 0.7863 0.7862  0.7902 0.7939    0
## glm_sig1 0.7835  0.7861 0.7885 0.7896  0.7939 0.7971    0
## glm_sig2 0.7841  0.7873 0.7907 0.7909  0.7946 0.7984    0
## glmnet   0.7792  0.7883 0.7904 0.7900  0.7914 0.7966    0
## rf       0.7683  0.7751 0.7767 0.7768  0.7791 0.7843    0
##
## Sens
##              Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
## glm_full 0.8804  0.8873 0.8982 0.8994  0.9127 0.9179    0
## glm_sig1 0.8861  0.8916 0.8997 0.9029  0.9147 0.9238    0
## glm_sig2 0.8883  0.9077 0.9099 0.9141  0.9258 0.9429    0
## glmnet   0.9055  0.9142 0.9246 0.9255  0.9391 0.9436    0
## rf       0.9688  0.9766 0.9836 0.9818  0.9856 0.9933    0
##
## Spec
##               Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
## glm_full 0.37360  0.3968 0.4237 0.4266  0.4548 0.4923    0
## glm_sig1 0.37460  0.3904 0.4212 0.4225  0.4494 0.4861    0
## glm_sig2 0.34210  0.3638 0.3982 0.3960  0.4170 0.4649    0
## glmnet   0.30190  0.3268 0.3592 0.3603  0.3917 0.4324    0
## rf       0.05521  0.1045 0.1373 0.1309  0.1547 0.1997    0
```

```r
dotplot(resamps, metric = 'ROC')
```



SELECT MODEL

- The final models were selected because they performed the best, are very simple and are highly intuitive
- For the linear regression model, it makes sense that the BLUEBOOK variable was the main (and only) factor as the cost of damage in an accident is related to the value of the car (excluding injuries, other property, etc.)
- For the logistic regression model, higher values for the following variables were associated with higher probability of an accident: KIDSDRIV, PARENT1, MSTATUS, TRAVTIME, Sports Car & SUV, CLM_FREQ, REVOKED, MVR_PTS
- Lower values for the following variables were associated with higher probability of accident: INCOME, HOME_VAL, BLUEBOOK, OLDCLAIM, Highly Urban
- After the final models were selected, we then re-fit the models to the entire data set (i.e. no cross validation) to ensure that we maximize use of all the available data
- The final logistic regression model is then used to predict the classes and probabilities
- Finally, the final linear regression model is used to predict the cost of damage for only those predicted accidents

```
final_linReg = train(TARGET_AMT ~ BLUEBOOK, data = insur_acc,
                     metric = 'RMSE', method = 'glm', preProcess = c('center', 'scale'),
```

```r
                      trControl = trainControl(verboseIter = FALSE))
summary(final_linReg)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -7757   -3083   -1541     295  101459
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5702.2      165.7  34.403  < 2e-16 ***
## BLUEBOOK        914.4      165.8   5.515  3.9e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 59148308)
##
##     Null deviance: 1.2903e+11  on 2152  degrees of freedom
## Residual deviance: 1.2723e+11  on 2151  degrees of freedom
## AIC: 44643
##
## Number of Fisher Scoring iterations: 2
```

```r
final_logReg = train(TARGET_FLAG ~ KIDSDRIV + INCOME + PARENT1 + HOME_VAL + MSTATUS +
                      TRAVTIME + CAR_USE + BLUEBOOK + TIF + CAR_TYPE +
                      OLDCLAIM + CLM_FREQ + REVOKED + MVR_PTS + URBANICITY, data = insur,
                      metric = 'ROC', method = 'glm', preProcess = c('center', 'scale'),
                      trControl = trainControl(summaryFunction = twoClassSummary, classProbs = TRUE, verb
                      na.action = na.exclude)
summary(final_logReg)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4802  -0.7381  -0.4165   0.6566   3.0520
##
## Coefficients:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)                         -1.40883    0.03449 -40.851  < 2e-16 ***
## KIDSDRIV                             0.21428    0.02790   7.680 1.59e-14 ***
## INCOME                              -0.34018    0.04067  -8.365  < 2e-16 ***
## PARENT1Yes                           0.16523    0.03140   5.261 1.43e-07 ***
## HOME_VAL                            -0.18563    0.04138  -4.486 7.25e-06 ***
## MSTATUSz_No                          0.20075    0.03821   5.253 1.49e-07 ***
## TRAVTIME                             0.23495    0.02952   7.958 1.75e-15 ***
## CAR_USEPrivate                      -0.43453    0.03343 -12.996  < 2e-16 ***
## BLUEBOOK                            -0.21301    0.03939  -5.408 6.39e-08 ***
## TIF                                 -0.22419    0.03010  -7.449 9.41e-14 ***
```

```
## `CAR_TYPEPanel Truck`              0.13557    0.03861   3.511 0.000446 ***
## CAR_TYPEPickup                     0.18284    0.03644   5.017 5.24e-07 ***
## `CAR_TYPESports Car`               0.29235    0.03304   8.849  < 2e-16 ***
## CAR_TYPEVan                        0.16559    0.03427   4.832 1.35e-06 ***
## CAR_TYPEz_SUV                      0.31487    0.03788   8.312  < 2e-16 ***
## OLDCLAIM                          -0.12042    0.03383  -3.560 0.000371 ***
## CLM_FREQ                           0.22060    0.03262   6.763 1.35e-11 ***
## REVOKEDYes                         0.29306    0.02951   9.932  < 2e-16 ***
## MVR_PTS                            0.25290    0.02887   8.760  < 2e-16 ***
## `URBANICITYz_Highly Rural/ Rural` -0.91146    0.04520 -20.166  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 9418  on 8160  degrees of freedom
## Residual deviance: 7430  on 8141  degrees of freedom
## AIC: 7470
##
## Number of Fisher Scoring iterations: 5
```

```
# import and cleanse test data
insur_test = read.csv('insurance-evaluation-data.csv', stringsAsFactors = TRUE)
# eliminate dollar signs and commas and convert to numbers
insur_test$INCOME = as.numeric(sub('\\$', '', sub('\\,', '', insur_test$INCOME)))
insur_test$HOME_VAL = as.numeric(sub('\\$', '', sub('\\,', '', insur_test$HOME_VAL)))
insur_test$BLUEBOOK = as.numeric(sub('\\$', '', sub('\\,', '', insur_test$BLUEBOOK)))
insur_test$OLDCLAIM = as.numeric(sub('\\$', '', sub('\\,', '', insur_test$OLDCLAIM)))
insur_test$AGE = impute_median(insur_test$AGE)
insur_test$YOJ = impute_median(insur_test$YOJ)
insur_test$INCOME = impute_median(insur_test$INCOME)
insur_test$HOME_VAL = impute_median(insur_test$HOME_VAL)
insur_test$CAR_AGE = impute_median(insur_test$CAR_AGE)

# predict classes and probabilities
pred_class = predict(final_logReg, newdata = insur_test)
pred_prob = predict(final_logReg, newdata = insur_test, type = 'prob')
insur_test$TARGET_FLAG = pred_class

# predict cost of damage
insur_test_acc = insur_test[insur_test$TARGET_FLAG == 'yes_crash',]
pred_cost = predict(final_linReg, newdata = insur_test_acc)
insur_test[insur_test$TARGET_FLAG == 'yes_crash', 'TARGET_AMT'] = pred_cost
insur_test[insur_test$TARGET_FLAG == 'no_crash', 'TARGET_AMT'] = 0

# finalize predictions
insur_test_classified = cbind(pred_prob, insur_test)
write.csv(insur_test_classified, 'insurance-evaluation-prediction.csv')

# the predictions for the evaluation data show less likelihood of accident compared to the training set
table(pred_class) / length(pred_class)
```

```
## pred_class
##  no_crash yes_crash
## 0.8435311 0.1564689
```

```r
# the predicted damage is relatively close to the average damge cost of the training set
mean(insur_test[insur_test$TARGET_FLAG == 'yes_crash', 'TARGET_AMT'])
```

```
## [1] 5439.619
```