BHao_HW5

DATA EXPLORATION

- There are many NAs scattered throughout the data set; we'll need to decide on how best to handle them
- First, let's take a look at the distribution of the TARGET variable
- Next, we only look at those observations where data is missing and review the TARGET variable distribution
- It appears that the distributions are mostly similar except in the case of the STARS variable, where there is a much greater proportion of 0 cases as compared to the other variables
- As such, we may be able to impute medians for the other variables, but in the case of the STARS variable, we'll set the missing data equal to 0 since there are no other zeros within this variable

```
library(dplyr)
library(ggplot2)
library(gridExtra)

wine = read.csv('wine-training-data.csv', stringsAsFactors = TRUE)
# drop index column
wine = subset(wine, select = -c(ï..INDEX))
str(wine)
```

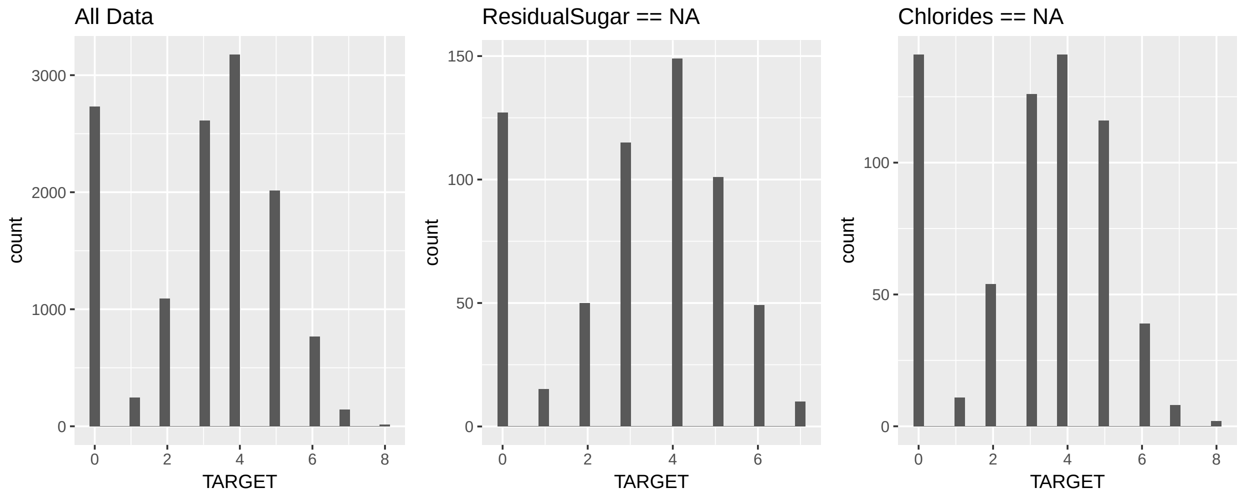
```
## 'data.frame':    12795 obs. of  15 variables:
##  $ TARGET            : int  3 3 5 3 4 0 0 4 3 6 ...
##  $ FixedAcidity      : num  3.2 4.5 7.1 5.7 8 11.3 7.7 6.5 14.8 5.5 ...
##  $ VolatileAcidity   : num  1.16 0.16 2.64 0.385 0.33 0.32 0.29 -1.22 0.27 -0.22 ...
##  $ CitricAcid        : num  -0.98 -0.81 -0.88 0.04 -1.26 0.59 -0.4 0.34 1.05 0.39 ...
##  $ ResidualSugar     : num  54.2 26.1 14.8 18.8 9.4 ...
##  $ Chlorides         : num  -0.567 -0.425 0.037 -0.425 NA 0.556 0.06 0.04 -0.007 -0.277 ...
##  $ FreeSulfurDioxide : num  NA 15 214 22 -167 -37 287 523 -213 62 ...
##  $ TotalSulfurDioxide: num  268 -327 142 115 108 15 156 551 NA 180 ...
##  $ Density           : num  0.993 1.028 0.995 0.996 0.995 ...
##  $ pH                : num  3.33 3.38 3.12 2.24 3.12 3.2 3.49 3.2 4.93 3.09 ...
##  $ Sulphates         : num  -0.59 0.7 0.48 1.83 1.77 1.29 1.21 NA 0.26 0.75 ...
##  $ Alcohol           : num  9.9 NA 22 6.2 13.7 15.4 10.3 11.6 15 12.6 ...
##  $ LabelAppeal       : int  0 -1 -1 -1 0 0 0 1 0 0 ...
##  $ AcidIndex         : int  8 7 8 6 9 11 8 7 6 8 ...
##  $ STARS             : int  2 3 3 1 2 NA NA 3 NA 4 ...
```

```
summary(wine)
```

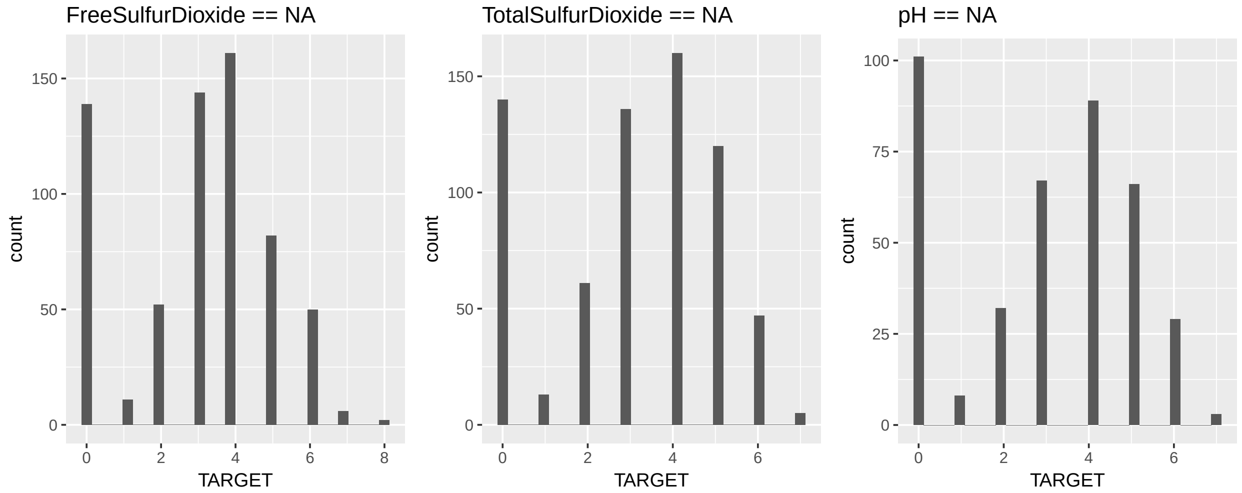
```
##      TARGET       FixedAcidity     VolatileAcidity     CitricAcid
##  Min.   :0.000   Min.   :-18.100   Min.   :-2.7900   Min.   :-3.2400
##  1st Qu.:2.000   1st Qu.:  5.200   1st Qu.: 0.1300   1st Qu.: 0.0300
##  Median :3.000   Median :  6.900   Median : 0.2800   Median : 0.3100
##  Mean   :3.029   Mean   :  7.076   Mean   : 0.3241   Mean   : 0.3084
##  3rd Qu.:4.000   3rd Qu.:  9.500   3rd Qu.: 0.6400   3rd Qu.: 0.5800
##  Max.   :8.000   Max.   : 34.400   Max.   : 3.6800   Max.   : 3.8600
##
```

```
##  ResidualSugar        Chlorides       FreeSulfurDioxide TotalSulfurDioxide
##  Min.   :-127.800   Min.   :-1.1710   Min.   :-555.00   Min.   :-823.0
##  1st Qu.:  -2.000   1st Qu.:-0.0310   1st Qu.:   0.00   1st Qu.:  27.0
##  Median :   3.900   Median : 0.0460   Median :  30.00   Median : 123.0
##  Mean   :   5.419   Mean   : 0.0548   Mean   :  30.85   Mean   : 120.7
##  3rd Qu.:  15.900   3rd Qu.: 0.1530   3rd Qu.:  70.00   3rd Qu.: 208.0
##  Max.   : 141.150   Max.   : 1.3510   Max.   : 623.00   Max.   :1057.0
##  NA's   :616        NA's   :638       NA's   :647       NA's   :682
##     Density             pH          Sulphates          Alcohol
##  Min.   :0.8881   Min.   :0.480   Min.   :-3.1300   Min.   :-4.70
##  1st Qu.:0.9877   1st Qu.:2.960   1st Qu.: 0.2800   1st Qu.: 9.00
##  Median :0.9945   Median :3.200   Median : 0.5000   Median :10.40
##  Mean   :0.9942   Mean   :3.208   Mean   : 0.5271   Mean   :10.49
##  3rd Qu.:1.0005   3rd Qu.:3.470   3rd Qu.: 0.8600   3rd Qu.:12.40
##  Max.   :1.0992   Max.   :6.130   Max.   : 4.2400   Max.   :26.50
##                   NA's   :395     NA's   :1210      NA's   :653
##   LabelAppeal          AcidIndex          STARS
##  Min.   :-2.000000   Min.   : 4.000   Min.   :1.000
##  1st Qu.:-1.000000   1st Qu.: 7.000   1st Qu.:1.000
##  Median : 0.000000   Median : 8.000   Median :2.000
##  Mean   :-0.009066   Mean   : 7.773   Mean   :2.042
##  3rd Qu.: 1.000000   3rd Qu.: 8.000   3rd Qu.:3.000
##  Max.   : 2.000000   Max.   :17.000   Max.   :4.000
##                                       NA's   :3359
```

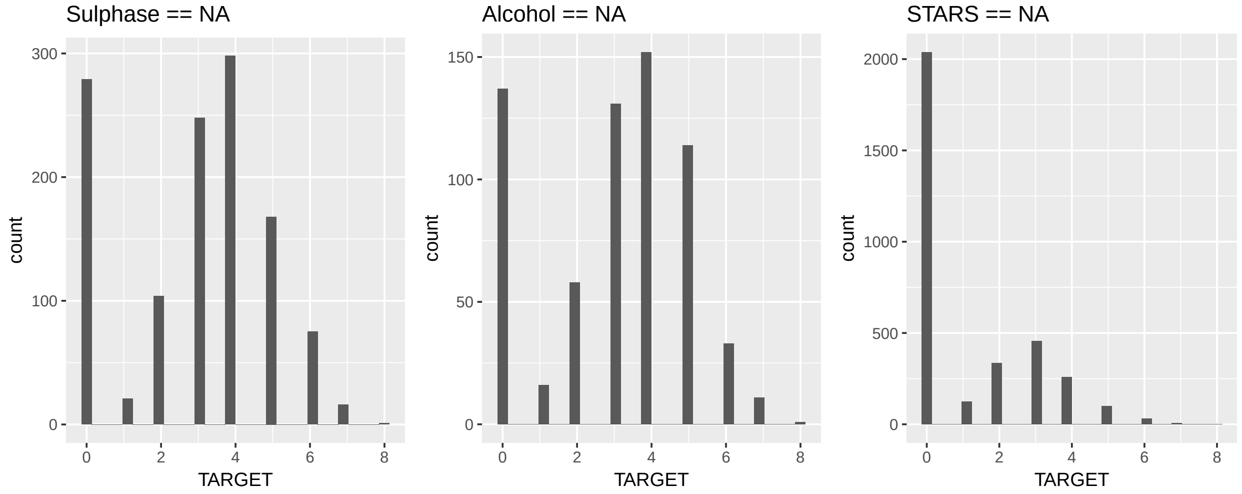
```r
p1 = wine %>% ggplot(aes(x = TARGET)) + geom_histogram() + ggtitle('All Data')
p2 = wine %>% filter(is.na(ResidualSugar)) %>% ggplot(aes(x = TARGET)) + geom_histogram() +
  ggtitle('ResidualSugar == NA')
p3 = wine %>% filter(is.na(Chlorides)) %>% ggplot(aes(x = TARGET)) + geom_histogram() +
  ggtitle('Chlorides == NA')
p4 = wine %>% filter(is.na(FreeSulfurDioxide)) %>% ggplot(aes(x = TARGET)) + geom_histogram() +
  ggtitle('FreeSulfurDioxide == NA')
p5 = wine %>% filter(is.na(TotalSulfurDioxide)) %>% ggplot(aes(x = TARGET)) + geom_histogram() +
  ggtitle('TotalSulfurDioxide == NA')
p6 = wine %>% filter(is.na(pH)) %>% ggplot(aes(x = TARGET)) + geom_histogram() +
  ggtitle('pH == NA')
p7 = wine %>% filter(is.na(Sulphates)) %>% ggplot(aes(x = TARGET)) + geom_histogram() +
  ggtitle('Sulphase == NA')
p8 = wine %>% filter(is.na(Alcohol)) %>% ggplot(aes(x = TARGET)) + geom_histogram() +
  ggtitle('Alcohol == NA')
p9 = wine %>% filter(is.na(STARS)) %>% ggplot(aes(x = TARGET)) + geom_histogram() +
  ggtitle('STARS == NA')

grid.arrange(p1, p2, p3, ncol = 3)
```



```r
grid.arrange(p4, p5, p6, ncol = 3)
```



```r
grid.arrange(p7, p8, p9, ncol = 3)
```



DATA PREPARATION

- We'll first impute medians for the non-STARS missing data
- We'll then set the NAs in STARS to 0
- Lastly, we'll add a flag to indicate STARS == 0

```
# median imputation
impute_median = function(x) replace(x, is.na(x), median(x, na.rm = TRUE))
wine$ResidualSugar = impute_median(wine$ResidualSugar)
wine$Chlorides = impute_median(wine$Chlorides)
wine$FreeSulfurDioxide = impute_median(wine$FreeSulfurDioxide)
wine$TotalSulfurDioxide = impute_median(wine$TotalSulfurDioxide)
wine$pH = impute_median(wine$pH)
wine$Sulphates = impute_median(wine$Sulphates)
wine$Alcohol = impute_median(wine$Alcohol)
wine[is.na(wine$STARS), 'STARS'] = 0
wine$STARS_FLAG = if_else(wine$STARS == 0, 'NoStars', 'Stars')
summary(wine)
```

```
##      TARGET       FixedAcidity     VolatileAcidity     CitricAcid
##  Min.   :0.000   Min.   :-18.100   Min.   :-2.7900   Min.   :-3.2400
##  1st Qu.:2.000   1st Qu.:  5.200   1st Qu.: 0.1300   1st Qu.: 0.0300
##  Median :3.000   Median :  6.900   Median : 0.2800   Median : 0.3100
##  Mean   :3.029   Mean   :  7.076   Mean   : 0.3241   Mean   : 0.3084
##  3rd Qu.:4.000   3rd Qu.:  9.500   3rd Qu.: 0.6400   3rd Qu.: 0.5800
##  Max.   :8.000   Max.   : 34.400   Max.   : 3.6800   Max.   : 3.8600
##  ResidualSugar        Chlorides        FreeSulfurDioxide
##  Min.   :-127.800   Min.   :-1.17100   Min.   :-555.0
##  1st Qu.:   0.900   1st Qu.: 0.00000   1st Qu.:   5.0
##  Median :   3.900   Median : 0.04600   Median :  30.0
##  Mean   :   5.346   Mean   : 0.05438   Mean   :  30.8
##  3rd Qu.:  14.900   3rd Qu.: 0.12800   3rd Qu.:  64.0
##  Max.   : 141.150   Max.   : 1.35100   Max.   : 623.0
##  TotalSulfurDioxide    Density             pH          Sulphates
##  Min.   :-823.0     Min.   :0.8881   Min.   :0.480   Min.   :-3.1300
##  1st Qu.:  34.0     1st Qu.:0.9877   1st Qu.:2.970   1st Qu.: 0.3400
##  Median : 123.0     Median :0.9945   Median :3.200   Median : 0.5000
##  Mean   : 120.8     Mean   :0.9942   Mean   :3.207   Mean   : 0.5245
##  3rd Qu.: 198.0     3rd Qu.:1.0005   3rd Qu.:3.450   3rd Qu.: 0.7700
##  Max.   :1057.0     Max.   :1.0992   Max.   :6.130   Max.   : 4.2400
##     Alcohol       LabelAppeal          AcidIndex          STARS
##  Min.   :-4.70   Min.   :-2.000000   Min.   : 4.000   Min.   :0.000
##  1st Qu.: 9.10   1st Qu.:-1.000000   1st Qu.: 7.000   1st Qu.:0.000
##  Median :10.40   Median : 0.000000   Median : 8.000   Median :1.000
##  Mean   :10.48   Mean   :-0.009066   Mean   : 7.773   Mean   :1.506
##  3rd Qu.:12.20   3rd Qu.: 1.000000   3rd Qu.: 8.000   3rd Qu.:2.000
##  Max.   :26.50   Max.   : 2.000000   Max.   :17.000   Max.   :4.000
##   STARS_FLAG
##  Length:12795
##  Class :character
##  Mode  :character
##
##
```

```
##
```

BUILD MODELS

- We'll first build a count regression model to estimate the number of cases sold
- 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, there does not appear to be much improvement as the model is simplified; still, for the sake of parsimony, we'll use the simplest version as our final model
- Note how well the rf model performed without manual tuning compared to the glmnet model which performed the poorest

```
library(caret)
library(caretEnsemble)

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(wine$TARGET, 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 ~ ., data = wine, method = 'glm', family = 'poisson',
                  preProcess = c('center', 'scale'), trControl = myControl)
summary(model_glm_full)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1709  -0.6521   0.0083   0.4525   3.7658
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)           9.725e-01  5.881e-03 165.349  < 2e-16 ***
## FixedAcidity          5.687e-05  5.178e-03   0.011  0.99124
## VolatileAcidity      -2.429e-02  5.111e-03  -4.752 2.01e-06 ***
## CitricAcid            4.836e-03  5.081e-03   0.952  0.34119
## ResidualSugar         2.235e-03  5.096e-03   0.439  0.66100
## Chlorides          -1.140e-02  5.114e-03  -2.229  0.02581 *  
## FreeSulfurDioxide   1.422e-02  5.083e-03   2.797  0.00516 ** 
## TotalSulfurDioxide  1.811e-02  5.134e-03   3.527  0.00042 ***
## Density            -7.365e-03  5.091e-03  -1.447  0.14800    
## pH                 -8.696e-03  5.116e-03  -1.700  0.08918 .  
## Sulphates          -1.037e-02  5.104e-03  -2.032  0.04217 *  
## Alcohol             1.245e-02  5.113e-03   2.434  0.01491 *  
## LabelAppeal         1.416e-01  5.460e-03  25.938  < 2e-16 ***
## AcidIndex          -1.069e-01  6.051e-03 -17.674  < 2e-16 ***
## STARS               2.228e-01  7.228e-03  30.824  < 2e-16 ***
## STARS_FLAGStars     2.850e-01  9.398e-03  30.323  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 22861  on 12794  degrees of freedom
## Residual deviance: 13767  on 12779  degrees of freedom
## AIC: 45741
## 
## Number of Fisher Scoring iterations: 6
```

```
# let's drop any statistically insignificant variables at 5%
model_glm_sig1 = train(TARGET ~ VolatileAcidity + Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +
                       pH + Sulphates + LabelAppeal + AcidIndex + STARS + STARS_FLAG,
                       data = wine, method = 'glm', family = 'poisson',
                       preProcess = c('center', 'scale'), trControl = myControl)
summary(model_glm_sig1)
```

```
## 
## Call:
## NULL
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.1724  -0.6551   0.0081   0.4569   3.7706  
## 
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)    
## (Intercept)          0.972555   0.005881 165.371  < 2e-16 ***
## VolatileAcidity     -0.024390   0.005111  -4.772 1.82e-06 ***
## Chlorides           -0.011865   0.005111  -2.321 0.020265 *  
## FreeSulfurDioxide    0.014001   0.005081   2.756 0.005858 ** 
## TotalSulfurDioxide   0.017760   0.005129   3.463 0.000534 ***
## pH                  -0.008838   0.005114  -1.728 0.083952 .  
## Sulphates           -0.010330   0.005102  -2.025 0.042888 *  
## LabelAppeal          0.141495   0.005460  25.916  < 2e-16 ***
## AcidIndex           -0.107613   0.005969 -18.028  < 2e-16 ***
## STARS                0.224168   0.007211  31.089  < 2e-16 ***
## STARS_FLAGStars      0.284458   0.009393  30.283  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
```

```
##     Null deviance: 22861  on 12794  degrees of freedom
## Residual deviance: 13776  on 12784  degrees of freedom
## AIC: 45740
##
## Number of Fisher Scoring iterations: 6
```

```
# let's again drop any additional statistically insigificant variables at 10%
model_glm_sig2 = train(TARGET ~ VolatileAcidity + FreeSulfurDioxide + TotalSulfurDioxide +
                       LabelAppeal + AcidIndex + STARS + STARS_FLAG,
                       data = wine,
                       method = 'glm', family = 'poisson',
                       preProcess = c('center', 'scale'), trControl = myControl)
summary(model_glm_sig2)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2018  -0.6526   0.0058   0.4546   3.7992
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)         0.972737   0.005880 165.428  < 2e-16 ***
## VolatileAcidity    -0.024506   0.005111  -4.795 1.63e-06 ***
## FreeSulfurDioxide   0.014080   0.005079   2.772 0.005569 **
## TotalSulfurDioxide  0.017837   0.005128   3.478 0.000504 ***
## LabelAppeal         0.141191   0.005459  25.865  < 2e-16 ***
## AcidIndex          -0.107352   0.005953 -18.033  < 2e-16 ***
## STARS               0.224540   0.007210  31.142  < 2e-16 ***
## STARS_FLAGStars     0.284924   0.009393  30.335  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 22861  on 12794  degrees of freedom
## Residual deviance: 13788  on 12787  degrees of freedom
## AIC: 45746
##
## Number of Fisher Scoring iterations: 6
```

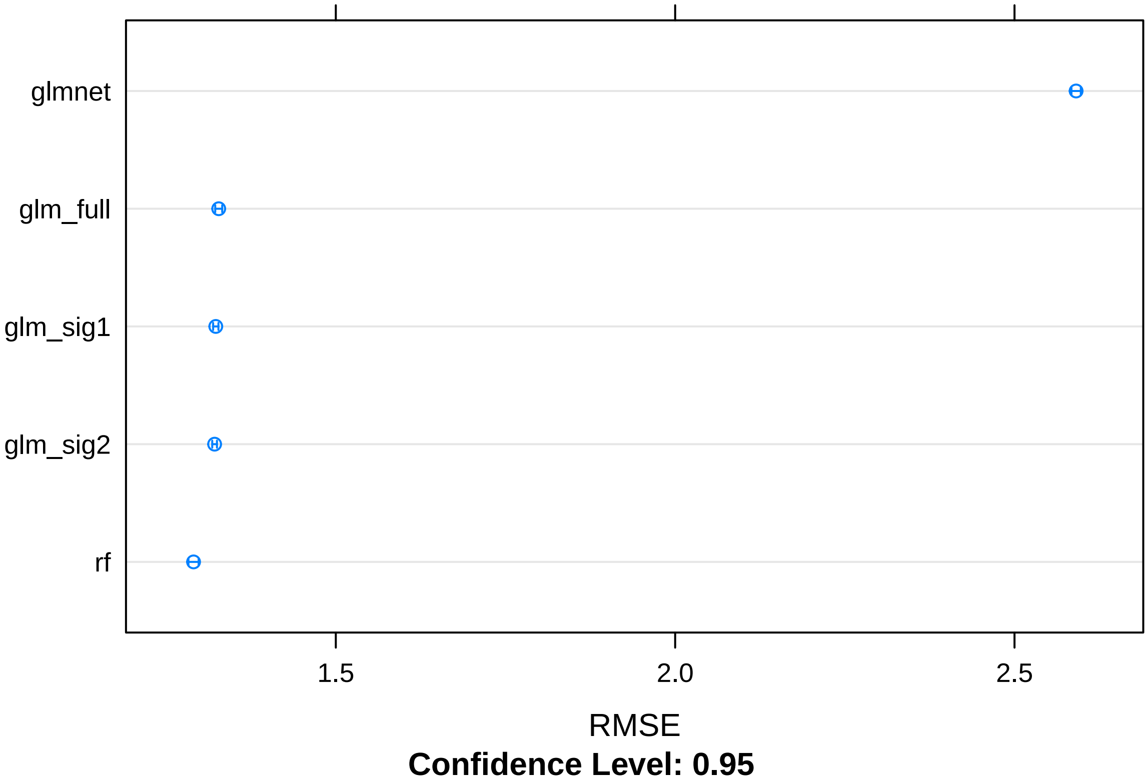
```
# let's try a glmnet model that combines ridge vs. lasso regression
# since it penalizes either or both magnitude and number of non-zero coefficients, it can be used for v
model_glmnet = train(TARGET ~ ., data = wine, method = 'glmnet', family = 'poisson',
                     preProcess = c('center',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

```
## 16 x 1 sparse Matrix of class "dgCMatrix"
##                                1
## (Intercept)         9.962303e-01
## FixedAcidity        .
## VolatileAcidity    -1.688800e-02
## CitricAcid          .
## ResidualSugar       .
## Chlorides          -3.611316e-03
## FreeSulfurDioxide   6.283616e-03
## TotalSulfurDioxide  9.393253e-03
## Density            -2.607628e-05
## pH                 -1.631423e-06
## Sulphates          -2.154344e-03
## Alcohol             5.013615e-03
## LabelAppeal         1.253206e-01
## AcidIndex          -9.141274e-02
## STARS               2.257313e-01
## STARS_FLAGStars     2.346651e-01
```

```
# let's also model using random forest just for fun
model_rf = train(TARGET ~ ., data = wine, method = 'ranger',
                 trControl = myControl)

# 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 1.317   1.323  1.327 1.327   1.333 1.337    0
## glm_sig1 1.315   1.320  1.323 1.323   1.326 1.334    0
## glm_sig2 1.313   1.319  1.322 1.321   1.324 1.328    0
## glmnet   2.574   2.588  2.589 2.591   2.596 2.607    0
## rf       1.276   1.283  1.289 1.290   1.297 1.310    0
##
## Rsquared
##            Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
## glm_full 0.5196  0.5232 0.5262 0.5262  0.5292 0.5325    0
## glm_sig1 0.5214  0.5281 0.5293 0.5291  0.5313 0.5331    0
## glm_sig2 0.5255  0.5291 0.5298 0.5302  0.5320 0.5347    0
## glmnet   0.5091  0.5118 0.5156 0.5173  0.5221 0.5289    0
## rf       0.5391  0.5468 0.5532 0.5521  0.5583 0.5608    0
```

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



SELECT MODEL

- The final models were selected because they performed the best, are very simple and are highly intuitive
- Since the variables were centered and scaled, we can interpret the coefficients on a more apples-to-apples basis:
  - STARS had the highest explanatory effect, with the more stars the more cases sold
  - Label appeal was the second most explanatory again with a higher appeal associated with more cases sold
  - AcidIndex was the third with less acid associated with more cases sold
- 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_model = train(TARGET ~ VolatileAcidity + FreeSulfurDioxide + TotalSulfurDioxide +
                     LabelAppeal + AcidIndex + STARS + STARS_FLAG,
                     data = wine,
                     method = 'glm', family = 'poisson',
                     preProcess = c('center', 'scale'),
                     trControl = trainControl(verboseIter = FALSE))
summary(final_model)
```

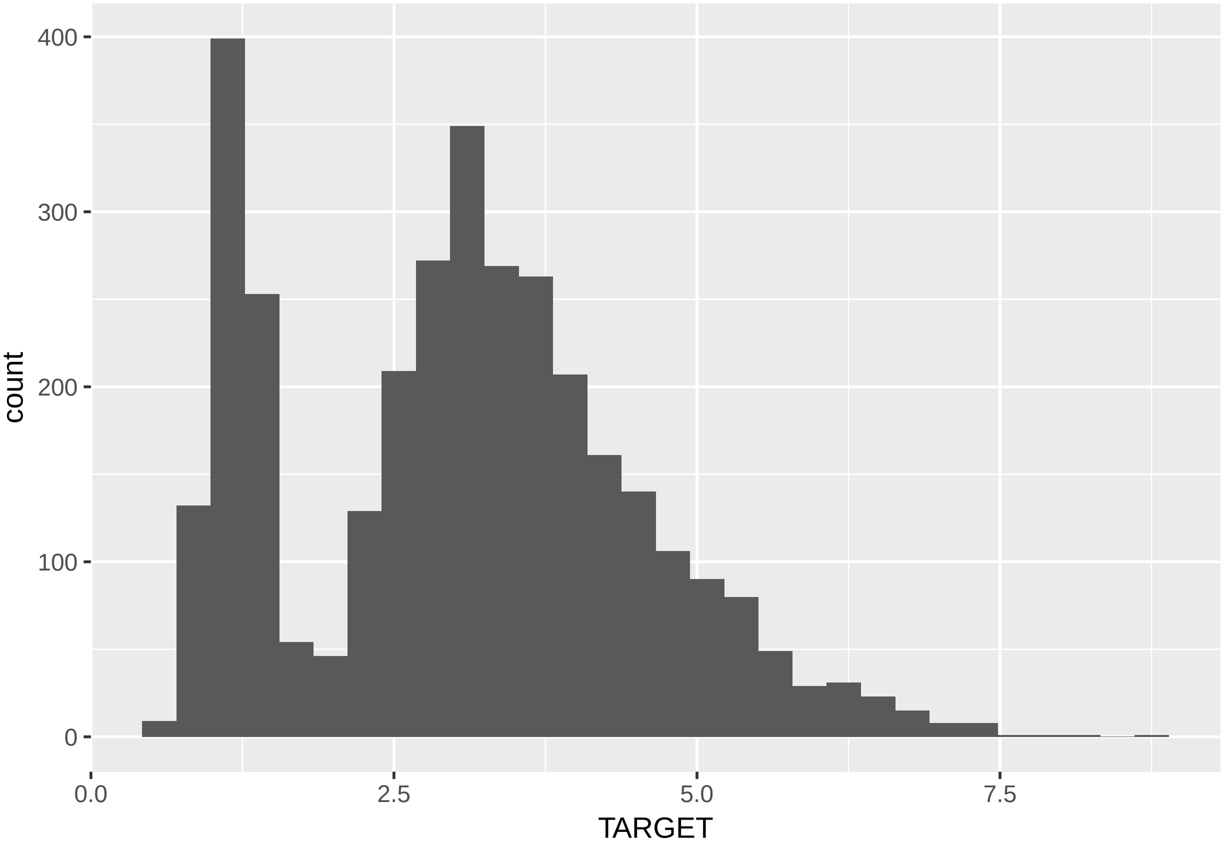
```
## 
## Call:
## NULL
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.2018  -0.6526   0.0058   0.4546   3.7992  
## 
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)    
## (Intercept)          0.972737   0.005880 165.428  < 2e-16 ***
## VolatileAcidity     -0.024506   0.005111  -4.795 1.63e-06 ***
## FreeSulfurDioxide    0.014080   0.005079   2.772 0.005569 ** 
## TotalSulfurDioxide   0.017837   0.005128   3.478 0.000504 ***
## LabelAppeal          0.141191   0.005459  25.865  < 2e-16 ***
## AcidIndex           -0.107352   0.005953 -18.033  < 2e-16 ***
## STARS                0.224540   0.007210  31.142  < 2e-16 ***
## STARS_FLAGStars      0.284924   0.009393  30.335  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for poisson family taken to be 1)
## 
##     Null deviance: 22861  on 12794  degrees of freedom
## Residual deviance: 13788  on 12787  degrees of freedom
## AIC: 45746
## 
## Number of Fisher Scoring iterations: 6
```

```
# import and cleanse test data
wine_test = read.csv('wine-evaluation-data.csv', stringsAsFactors = TRUE)
wine_test$ResidualSugar = impute_median(wine_test$ResidualSugar)
wine_test$Chlorides = impute_median(wine_test$Chlorides)
wine_test$FreeSulfurDioxide = impute_median(wine_test$FreeSulfurDioxide)
wine_test$TotalSulfurDioxide = impute_median(wine_test$TotalSulfurDioxide)
wine_test$pH = impute_median(wine_test$pH)
wine_test$Sulphates = impute_median(wine_test$Sulphates)
wine_test$Alcohol = impute_median(wine_test$Alcohol)
wine_test[is.na(wine_test$STARS), 'STARS'] = 0
wine_test$STARS_FLAG = if_else(wine_test$STARS == 0, 'NoStars', 'Stars')
summary(wine_test)
```

```
##        IN          TARGET         FixedAcidity     VolatileAcidity  
##  Min.   :    3   Mode:logical   Min.   :-18.200   Min.   :-2.8300  
##  1st Qu.: 4018   NA's:3335      1st Qu.:  5.200   1st Qu.: 0.0800  
##  Median : 7906                  Median :  6.900   Median : 0.2800  
##  Mean   : 8048                  Mean   :  6.864   Mean   : 0.3103  
##  3rd Qu.:12061                  3rd Qu.:  9.000   3rd Qu.: 0.6300  
##  Max.   :16130                  Max.   : 33.500   Max.   : 3.6100  
##    CitricAcid      ResidualSugar        Chlorides        FreeSulfurDioxide
##  Min.   :-3.1200   Min.   :-128.300   Min.   :-1.15000   Min.   :-563.00  
##  1st Qu.: 0.0000   1st Qu.:   0.500   1st Qu.: 0.02400   1st Qu.:   5.00  
##  Median : 0.3100   Median :   3.600   Median : 0.04700   Median :  30.00  
##  Mean   : 0.3124   Mean   :   5.233   Mean   : 0.06083   Mean   :  34.72  
##  3rd Qu.: 0.6050   3rd Qu.:  15.525   3rd Qu.: 0.14350   3rd Qu.:  70.00  
##  Max.   : 3.7600   Max.   : 145.400   Max.   : 1.26300   Max.   : 617.00  
##  TotalSulfurDioxide    Density             pH           Sulphates      
##  Min.   :-769.0     Min.   :0.8898   Min.   :0.600   Min.   :-3.0700  
##  1st Qu.:  32.0     1st Qu.:0.9883   1st Qu.:2.990   1st Qu.: 0.3600  
##  Median : 124.0     Median :0.9946   Median :3.210   Median : 0.5000  
##  Mean   : 123.4     Mean   :0.9947   Mean   :3.236   Mean   : 0.5314  
##  3rd Qu.: 201.0     3rd Qu.:1.0005   3rd Qu.:3.460   3rd Qu.: 0.7550  
##  Max.   :1004.0     Max.   :1.0998   Max.   :6.210   Max.   : 4.1800  
##     Alcohol       LabelAppeal          AcidIndex          STARS      
##  Min.   :-4.20   Min.   :-2.00000   Min.   : 5.000   Min.   :0.000  
##  1st Qu.: 9.10   1st Qu.:-1.00000   1st Qu.: 7.000   1st Qu.:0.000  
##  Median :10.40   Median : 0.00000   Median : 8.000   Median :1.000  
##  Mean   :10.57   Mean   : 0.01349   Mean   : 7.748   Mean   :1.526  
##  3rd Qu.:12.40   3rd Qu.: 1.00000   3rd Qu.: 8.000   3rd Qu.:2.000  
##  Max.   :25.60   Max.   : 2.00000   Max.   :17.000   Max.   :4.000  
##   STARS_FLAG       
##  Length:3335       
##  Class :character  
##  Mode  :character  
##                    
##                    
## 
```

```r
# predict cases sold
pred = predict(final_model, newdata = wine_test)
wine_test$TARGET = pred

write.csv(wine_test, 'wine-evaluation-prediction.csv')

# check the distribution of predicted cases sold
wine_test %>% ggplot(aes(x = TARGET)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


400
300
200
100
0
count
0.0
2.5
5.0
7.5
TARGET