Stephen Woods

Wine Sale Prediction

Spring 2022

Summary

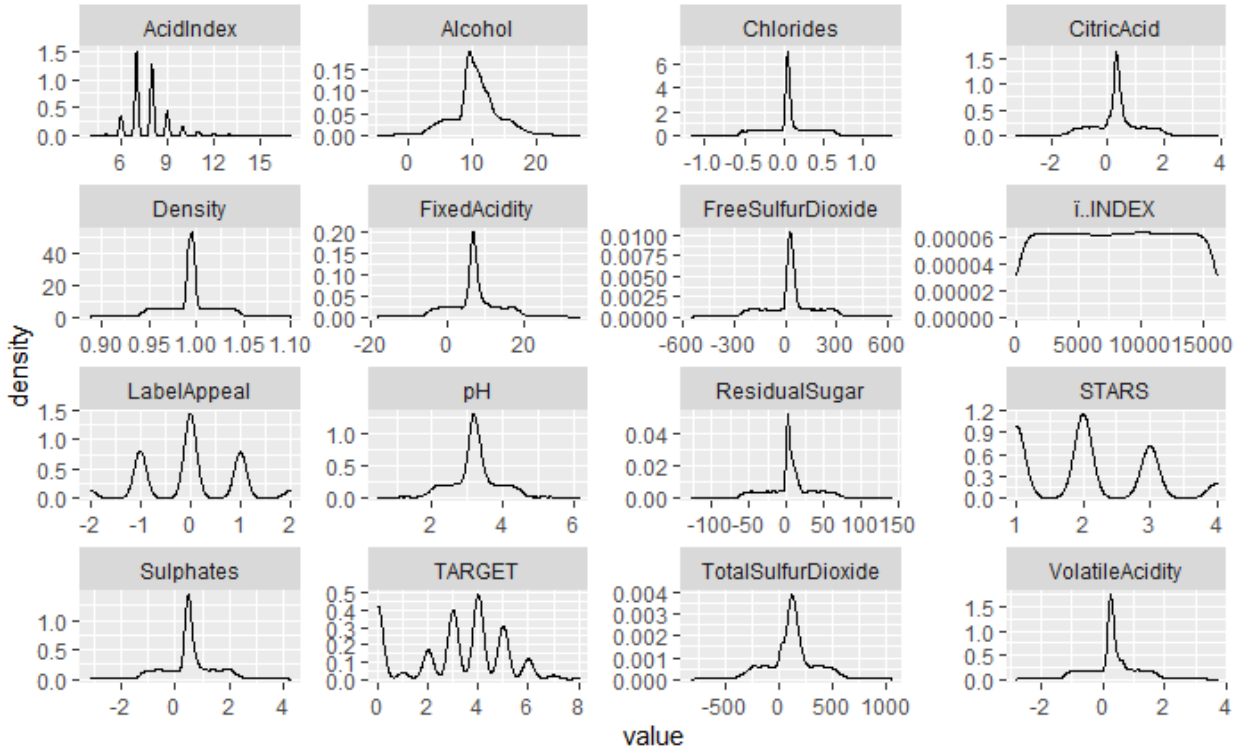
This notebook features EDA and logistic regression on a dataset containing chemical properties and ratings for around 12,000 commercially available wines.

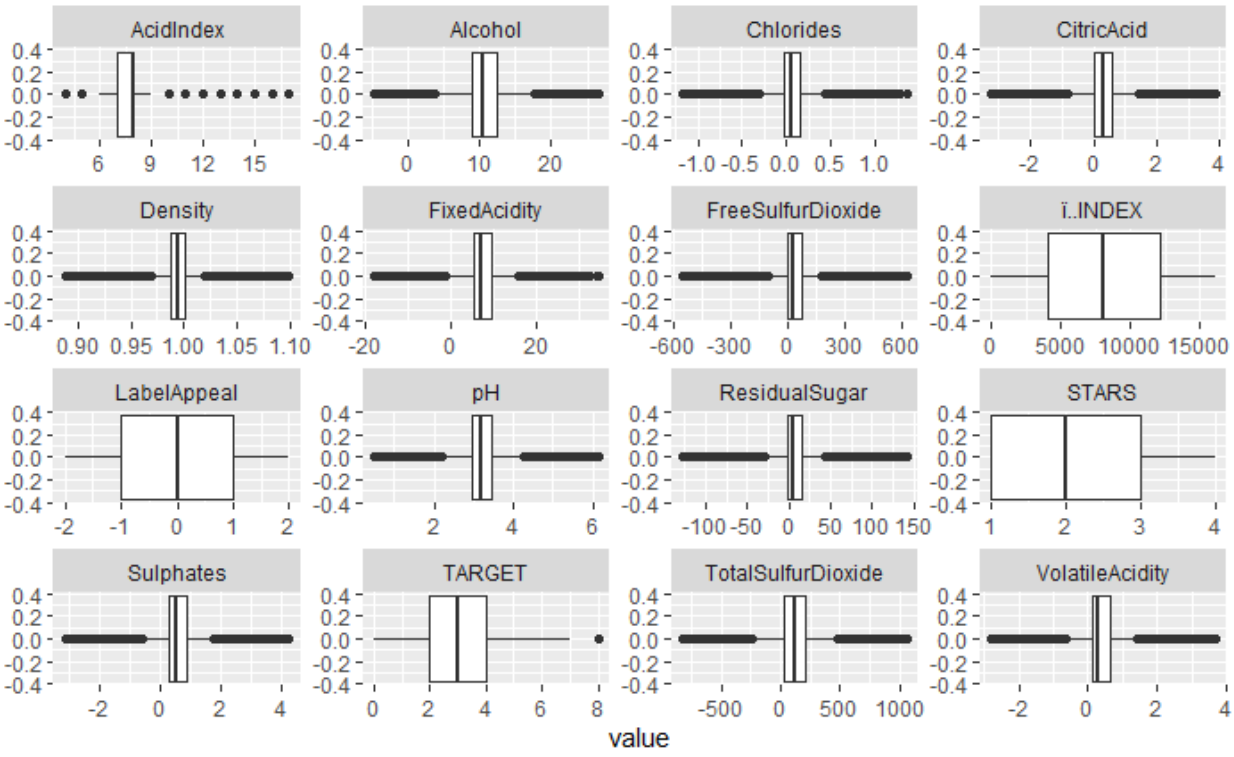
Section 1 – EDA

The training dataset contains 16 variables with 12,795 observations: 'TARGET' is the dependent variable and there are 14 independent variables measuring the chemical properties and rating for a given wine. There is one index column.

The dataset contains eight fields with between 395 and 3,359 NAs. Altogether, there are 8,200 rows with nonfinite values.

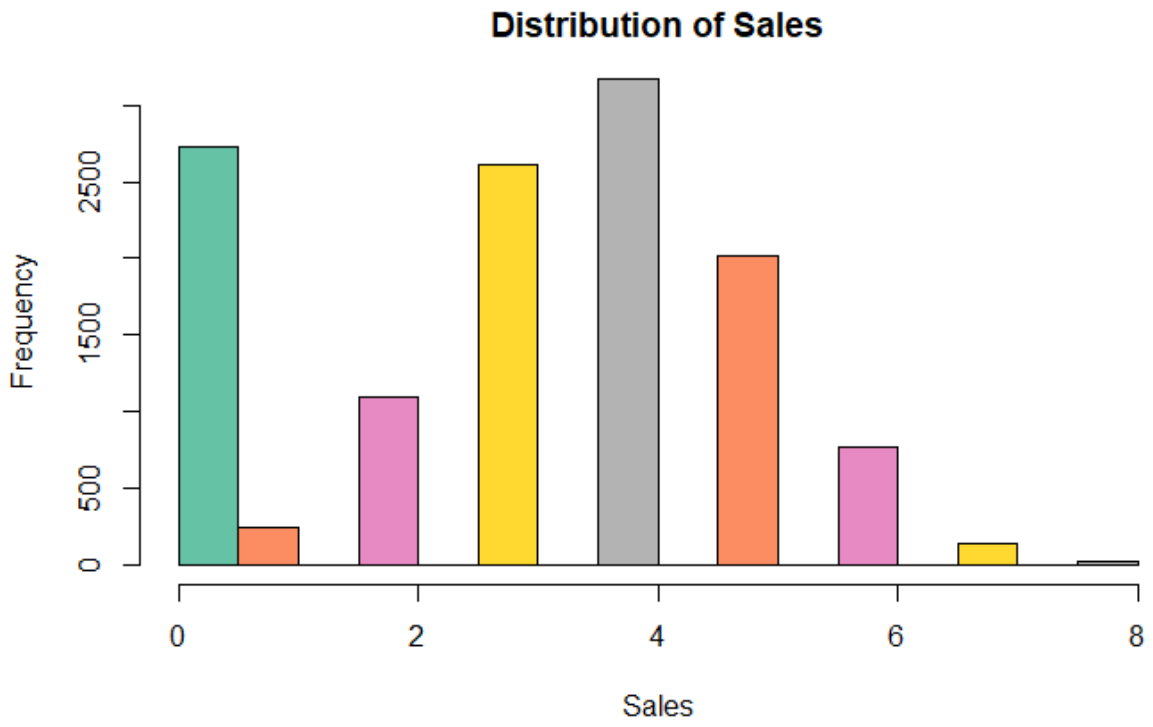
Below shows a density plot and box plot for all 16 variables:



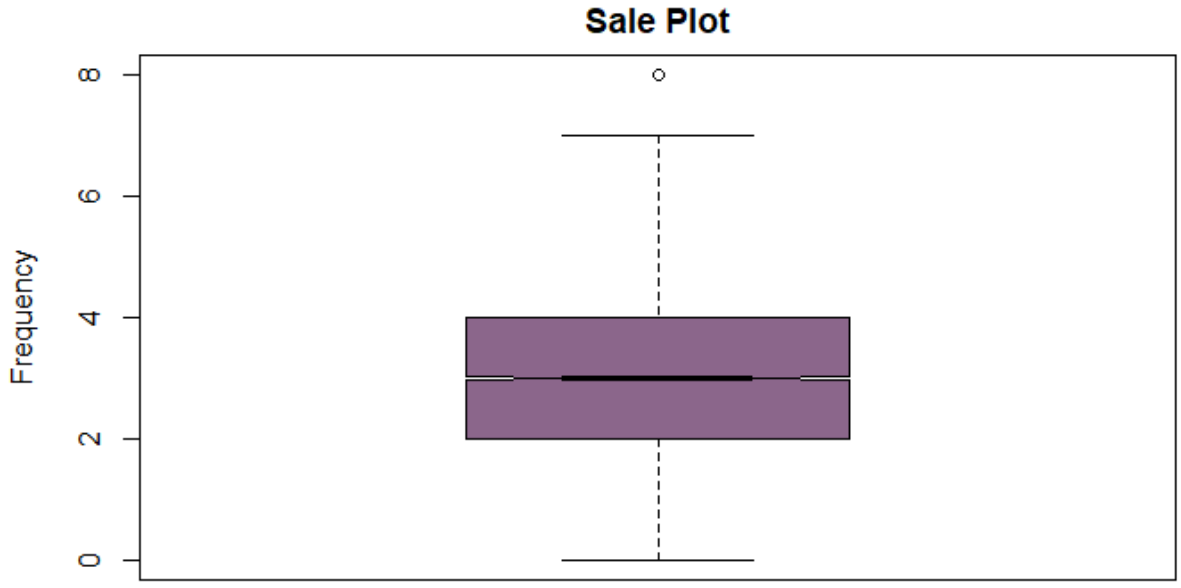


Note most of the distributions do not follow that of a normal distribution. Also note the large number of outliers in every column but TARGET, STARS, and LabelAppeal. These three columns also each have a density plot that follows that of a roller coaster or multi modal distribution.

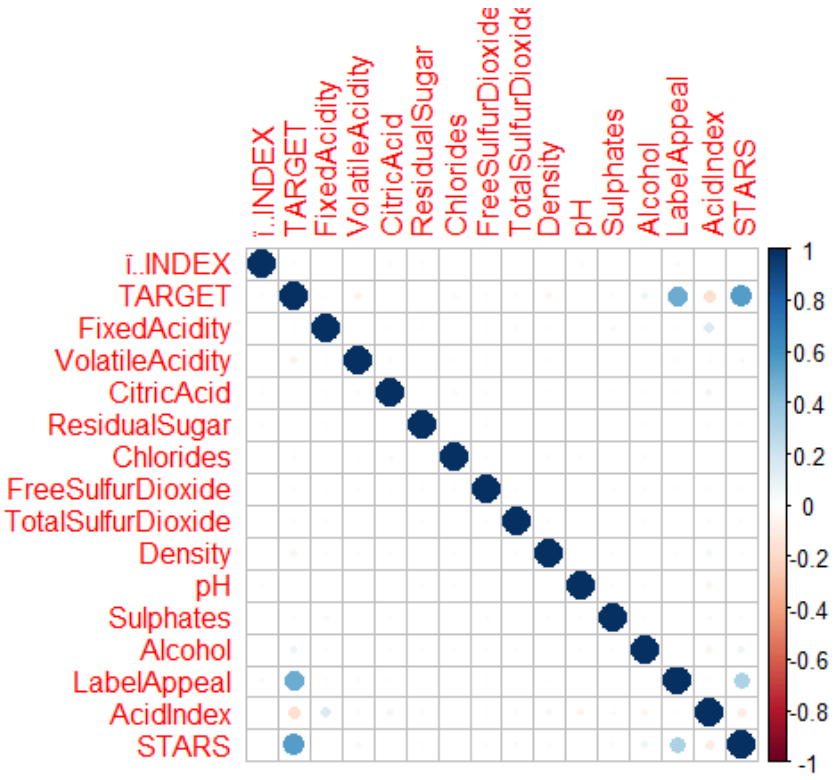
Looking closer at the TARGET variable, we see that 4 wine sales is the most common, followed by zero wine sales:



Further, there is only one outlier hitting 8 wine sales:



Next, correlations were examined. LabelAppeal, AcidIndex, and STARS variables all show a significant correlation with TARGET. Note there is some collinearity among these three variables:

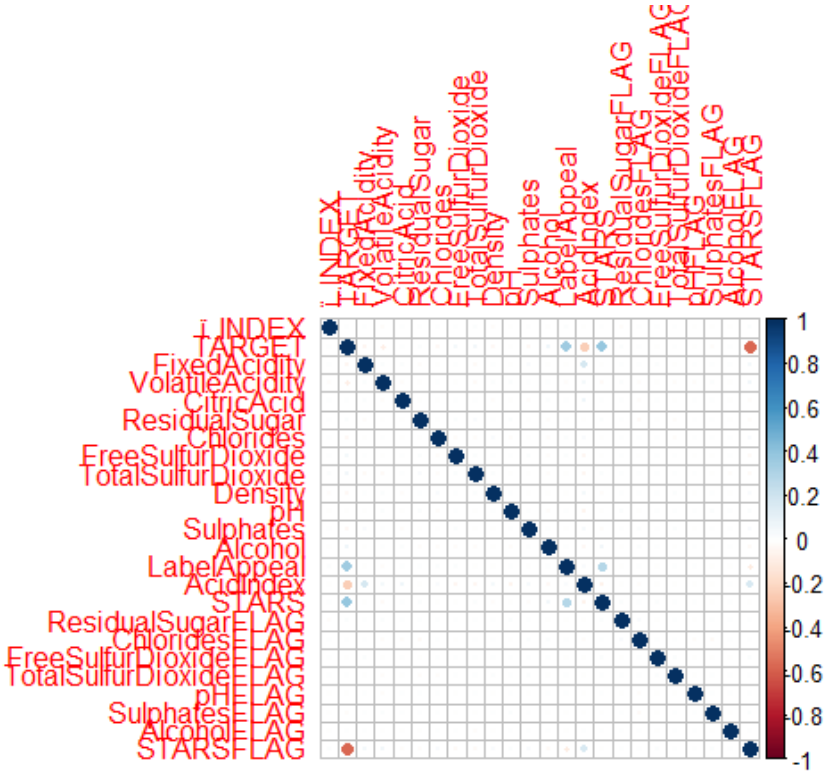


Section 2 – Data Preparation

1) First, 'flag' dummy columns were created for each of the eight variables with null values. These binary columns indicate whether an NA existed (1 = YES) in the original row for the given variable.

```r
# Flag Missing Values with new Dummy Columns (1 = missing value)
mydata$ResidualSugarFLAG <- ifelse(is.na(mydata$ResidualSugar), 1, 0)
mydata$ChloridesFLAG <- ifelse(is.na(mydata$Chlorides), 1, 0)
mydata$FreeSulfurDioxideFLAG <- ifelse(is.na(mydata$FreeSulfurDioxide), 1, 0)
mydata$TotalSulfurDioxideFLAG <- ifelse(is.na(mydata$TotalSulfurDioxide), 1, 0)
mydata$pHFLAG <- ifelse(is.na(mydata$pH), 1, 0)
mydata$SulphatesFLAG <- ifelse(is.na(mydata$Sulphates), 1, 0)
mydata$AlcoholFLAG <- ifelse(is.na(mydata$Alcohol), 1, 0)
mydata$STARSFLAG <- ifelse(is.na(mydata$STARS), 1, 0)
```

Note further EDA showed the 'STARSFLAG' column has the strongest correlation with TARGET of all potential predictor variables:



2) Next, all NA's were imputed with the mean value of a given column. Columns with NA's did not show a strong relationship with the TARGET variable, so imputing with means should suffice.

```
#Impute with Mean

for(i in 1:ncol(mydata)) {
  mydata[ , i][is.na(mydata[ , i])] <- mean(mydata[ , i], na.rm=TRUE)
}

names(which(colSums(is.na(mydata))>0))
```

3) Finally, a binary indicator 'TARGET_BIN' variable was created to indicate whether any sales were made (1 = SALES, 0 = NO SALES), as that is what we will be predicting.

```
#create yes/no column for whether sale was made, this will be the new target column

mydata$TARGET_BIN <- ifelse(mydata$TARGET == 0, 0, 1)
```

Steps 1 and 2 were also performed on the test dataset.

Section 3 – Model Building

First, a linear model was fit with predictors showing significant correlations with TARGET (shown below), featuring an adjusted R-Square of 0.3894. Then, a model with all independent variables (excluding index columns) was fit, featuring an adjusted R-Square of .3958. Note adding additional predictors drove a small change in model performance.

Next, logistic models were fitted to better fit the binary outcome, IE the probability of making any number of sales, we are trying to capture. The first logistic model, 'logmodel1', features the same four predictors as the first linear model, shown below. Logmodel1 achieves an AIC of 7763, accuracy of 85.53%, sensitivity of 90.95%, and precision of 90.68%.

```
#model with four variables
logmodel1 <- Logit(TARGET_BIN ~ LabelAppeal + AcidIndex + STARS + STARSFLAG, data = mydata)
logmodel1
```

```
Probability threshold for predicting : 0.5

                  Baseline         Predicted
-------------------------------------------------
                 Total  %Tot        0      1  %Correct
-------------------------------------------------------
            1    10061  78.6      911   9150     90.9
TARGET_BIN  0     2734  21.4     1794    940     65.6
-------------------------------------------------------
          Total  12795                           85.5

Accuracy: 85.53
Sensitivity: 90.95
Precision: 90.68


Call:  glm(formula = my_formula, family = "binomial", data = data)

Coefficients:
(Intercept)  LabelAppeal    AcidIndex        STARS    STARSFLAG
     1.8697      -0.4636      -0.3933       2.5541      -4.4693

Degrees of Freedom: 12794 Total (i.e. Null);  12790 Residual
Null Deviance:      13280
Residual Deviance: 7753         AIC: 7763
```

Below shows the significance of each model predictor:

```
   BASIC ANALYSIS

 Estimated Model for the Logit of Reference Group Membership

             Estimate    Std Err  z-value  p-value   Lower 95%   Upper 95%
 (Intercept)   1.8697     0.2156    8.671    0.000      1.4471      2.2923
 LabelAppeal  -0.4636     0.0330  -14.064    0.000     -0.5283     -0.3990
   AcidIndex  -0.3933     0.0211  -18.603    0.000     -0.4347     -0.3519
       STARS   2.5541     0.1116   22.885    0.000      2.3354      2.7729
   STARSFLAG  -4.4693     0.1149  -38.883    0.000     -4.6946     -4.2441


 Odds ratios and confidence intervals

             Odds Ratio   Lower 95%   Upper 95%
 (Intercept)      6.4864      4.2507      9.8980
 LabelAppeal      0.6290      0.5896      0.6710
   AcidIndex      0.6748      0.6474      0.7034
       STARS     12.8598     10.3332     16.0042
   STARSFLAG      0.0115      0.0091      0.0143
```

LabelAppeal has a coefficient estimate of -0.4636, which translates to a 37.10% percent change in the odds ratio for every one unit change in x. For each one unit increase in label appeal category, the odds of a wine being sold decreases by 37.10%. High scores on label appeal indicate consumers found the wine bottle label appealing, so I would think odds would increase as label appeal increases.

For each one unit increase in the AcidIndex category, the odds of a wine being sold decreases by 32.52%. Wine drinkers must not like wines with high acidity.

For each one unit increase in the STARS category, the odds of a wine being sold increases by 11,859.72%. For the STARSFLAG category, the odds of a wine being sold that was missing a starz value is 98.85% lower than a wine being sold that was not missing the STARS value. The STARS variable represents the expert wine rating with 1 = poor and 4 = excellent, so it makes sense that wines with higher ratings will be more likely to sell. Similarly, wines with no ratings are less likely to sell, maybe because consumers are hesitant to buy a wine with no ratings.

A second model was fit with any predictor that featured a 'three star/***' significance level from the linear regression model with all variables. This model features an AIC of 7674, accuracy of 85.92%, sensitivity of 92.10%, and precision of 90.21%.

```
logmodel2 <- Logit(TARGET_BIN ~ LabelAppeal + AcidIndex + STARS + STARSFLAG + VolatileAcidity + TotalSulfurDioxide + pH, data = mydata)
logmodel2
```

```
Probability threshold for predicting : 0.5

                  Baseline         Predicted
-----------------------------------------------------
                 Total  %Tot        0      1  %Correct
-----------------------------------------------------
            1    10061  78.6      795   9266     92.1
TARGET_BIN  0     2734  21.4     1728   1006     63.2
-----------------------------------------------------
          Total  12795                          85.9

Accuracy: 85.92
Sensitivity: 92.10
Precision: 90.21


Call:  glm(formula = my_formula, family = "binomial", data = data)

Coefficients:
      (Intercept)         LabelAppeal           AcidIndex               STARS
STARSFLAG     VolatileAcidity
         2.409002           -0.463481           -0.391252            2.550047
-4.474957           -0.182317
TotalSulfurDioxide                  pH
         0.000871           -0.178287

Degrees of Freedom: 12794 Total (i.e. Null);  12787 Residual
Null Deviance:      13280
Residual Deviance: 7658         AIC: 7674
```

Below are the coefficients of this model:

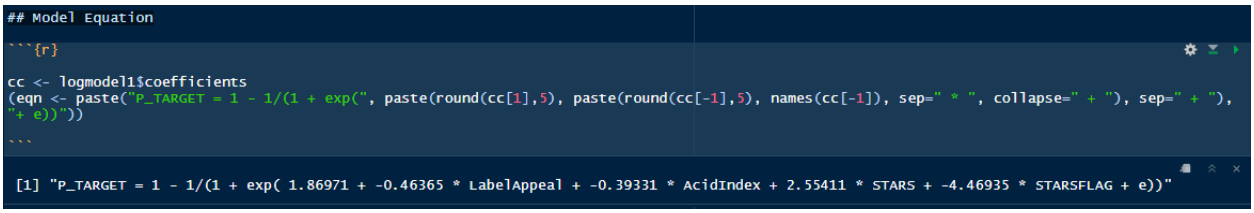
```
   BASIC ANALYSIS

Estimated Model for the Logit of Reference Group Membership

                   Estimate    Std Err  z-value  p-value   Lower 95%   Upper 95%
       (Intercept)   2.4090     0.2646    9.105    0.000      1.8905      2.9276
      LabelAppeal   -0.4635     0.0332  -13.950    0.000     -0.5286     -0.3984
        AcidIndex   -0.3913     0.0213  -18.372    0.000     -0.4330     -0.3495
            STARS    2.5500     0.1118   22.805    0.000      2.3309      2.7692
        STARSFLAG   -4.4750     0.1153  -38.808    0.000     -4.7010     -4.2490
   VolatileAcidity  -0.1823     0.0364   -5.009    0.000     -0.2537     -0.1110
TotalSulfurDioxide   0.0009     0.0001    6.887    0.000      0.0006      0.0011
                pH  -0.1783     0.0425   -4.197    0.000     -0.2615     -0.0950


Odds ratios and confidence intervals

                   Odds Ratio   Lower 95%   Upper 95%
       (Intercept)    11.1229      6.6224     18.6819
      LabelAppeal     0.6291      0.5894      0.6714
        AcidIndex     0.6762      0.6486      0.7050
            STARS    12.8077     10.2870     15.9460
        STARSFLAG     0.0114      0.0091      0.0143
   VolatileAcidity    0.8333      0.7760      0.8950
TotalSulfurDioxide    1.0009      1.0006      1.0011
                pH    0.8367      0.7699      0.9093
```

Section 4 – Model Selection

I selected logmodel1 as my final model. While logmodel1 has a higher AIC score than logmodel2, it has three fewer predictors and is therefore simpler. The small decrease in AIC score did not justify including more predictors in the model, and I was skeptical logmodel2 would have better performance on test data. Further, logmodel2 has a slightly lower precision score than logmodel1 (90.21 vs 90.68). I do not feel strongly about this decision as both models would probably reach similar results on test data, but I chose to err on the side of simplicity.

Section 5 – Model Equation

The below code was used to generate my model equation:



P_TARGET = 1 - 1/(1 + exp(1.86971 + -0.46365 * LabelAppeal + -0.39331 * AcidIndex + 2.55411 * STARS + -4.46935 * STARSFLAG + e))

Steps to clean the test data set, as discussed earlier, are shown below:

1) A 'flag' dummy column was created for the STARZ column to indicate whether an NA existed (1 = YES) in the original row for the given variable. Other flag columns were not created because I only used the STARSFLAG column in my model.

```
test$STARSFLAG <- ifelse(is.na(test$STARS), 1, 0)
```

2) Next, all NA's were imputed with the mean value of a given column.

```
for (i in 1:ncol(test)) {
  test[ , i][is.na(test[ , i])] <- mean(test[ , i], na.rm=TRUE)
}
```