DATA 621 - Homework 5

Fall 2020 - Business Analytics and Data Mining

Mohamed Thasleem, Kalikul Zaman

12/24/2020

Contents

Introduction

In this homework assignment, we will be exploring, analyzing and model a data set containing information on approximately 12,000 commercially available wines. The variables are mostly related to the chemical properties of the wine being sold. The response variable is the number of sample cases of wine that were purchased by wine distribution companies after sampling a wine. These cases would be used to provide tasting samples to restaurants and wine stores around the United States. The more sample cases purchased, the more likely is a wine to be sold at a high end restaurant. A large wine manufacturer is studying the data in order to predict the number of wine cases ordered based upon the wine characteristics. If the wine manufacturer can predict the number of cases, then that manufacturer will be able to adjust their wine offering to maximize sales.

1. Data Download

```r
# download data
path <- "https://raw.githubusercontent.com/mohamedthasleem/DATA621/master/HW5"
df <- read.csv(paste0(path,"/wine-training-data.csv"),header = TRUE)
eval <- read.csv(paste0(path,"/wine-evaluation-data.csv"),header = TRUE)
```

2. Data Exploration

Previewing the data, We will first look at the summary statistics for the data

```
head(df)
```

```
##   ï..INDEX TARGET FixedAcidity VolatileAcidity CitricAcid ResidualSugar
## 1        1      3          3.2           1.160      -0.98          54.2
## 2        2      3          4.5           0.160      -0.81          26.1
## 3        4      5          7.1           2.640      -0.88          14.8
## 4        5      3          5.7           0.385       0.04          18.8
## 5        6      4          8.0           0.330      -1.26           9.4
## 6        7      0         11.3           0.320       0.59           2.2
##   Chlorides FreeSulfurDioxide TotalSulfurDioxide Density   pH Sulphates Alcohol
## 1    -0.567                NA                268 0.99280 3.33     -0.59     9.9
## 2    -0.425                15               -327 1.02792 3.38      0.70      NA
## 3     0.037               214                142 0.99518 3.12      0.48    22.0
## 4    -0.425                22                115 0.99640 2.24      1.83     6.2
## 5        NA              -167                108 0.99457 3.12      1.77    13.7
## 6     0.556               -37                 15 0.99940 3.20      1.29    15.4
##   LabelAppeal AcidIndex STARS
## 1           0         8     2
## 2          -1         7     3
## 3          -1         8     3
## 4          -1         6     1
## 5           0         9     2
## 6           0        11    NA
```

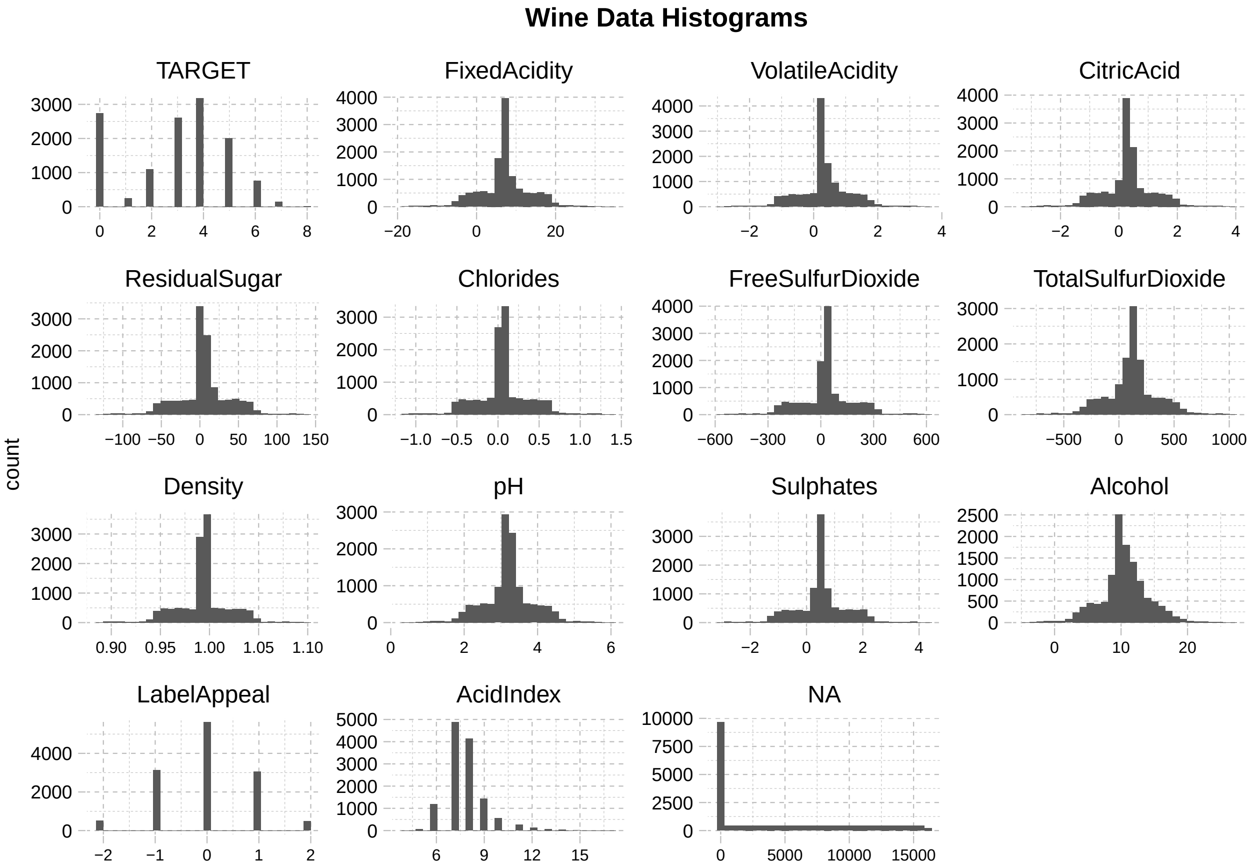
```
glimpse(df)
```

```
## Rows: 12,795
## Columns: 16
## $ ï..INDEX           <int> 1, 2, 4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17,...
## $ TARGET             <int> 3, 3, 5, 3, 4, 0, 0, 4, 3, 6, 0, 4, 3, 7, 4, 0, ...
## $ FixedAcidity       <dbl> 3.2, 4.5, 7.1, 5.7, 8.0, 11.3, 7.7, 6.5, 14.8, 5...
## $ VolatileAcidity    <dbl> 1.160, 0.160, 2.640, 0.385, 0.330, 0.320, 0.290,...
## $ CitricAcid         <dbl> -0.98, -0.81, -0.88, 0.04, -1.26, 0.59, -0.40, 0...
## $ ResidualSugar      <dbl> 54.20, 26.10, 14.80, 18.80, 9.40, 2.20, 21.50, 1...
## $ Chlorides          <dbl> -0.567, -0.425, 0.037, -0.425, NA, 0.556, 0.060,...
## $ FreeSulfurDioxide  <dbl> NA, 15, 214, 22, -167, -37, 287, 523, -213, 62, ...
## $ TotalSulfurDioxide <dbl> 268, -327, 142, 115, 108, 15, 156, 551, NA, 180,...
## $ Density            <dbl> 0.99280, 1.02792, 0.99518, 0.99640, 0.99457, 0.9...
## $ pH                 <dbl> 3.33, 3.38, 3.12, 2.24, 3.12, 3.20, 3.49, 3.20, ...
## $ Sulphates          <dbl> -0.59, 0.70, 0.48, 1.83, 1.77, 1.29, 1.21, NA, 0...
## $ Alcohol            <dbl> 9.9, NA, 22.0, 6.2, 13.7, 15.4, 10.3, 11.6, 15.0...
## $ LabelAppeal        <int> 0, -1, -1, -1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 2, 0, ...
## $ AcidIndex          <int> 8, 7, 8, 6, 9, 11, 8, 7, 6, 8, 5, 10, 7, 8, 9, 8...
## $ STARS              <int> 2, 3, 3, 1, 2, NA, NA, 3, NA, 4, 1, 2, 2, 3, NA,...
```

```
summary(df)
```

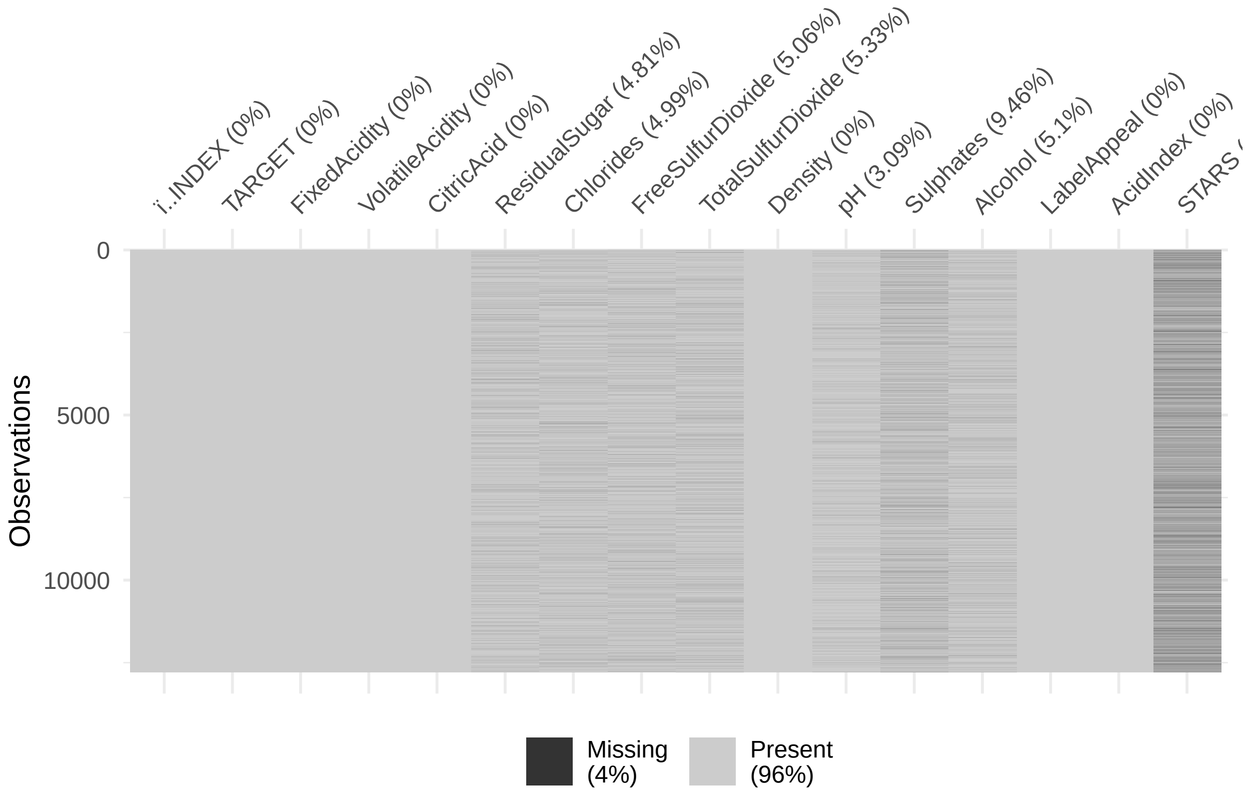
```
##     ï..INDEX          TARGET       FixedAcidity     VolatileAcidity
##  Min.   :    1   Min.   :0.000   Min.   :-18.100   Min.   :-2.7900
##  1st Qu.: 4038   1st Qu.:2.000   1st Qu.:  5.200   1st Qu.: 0.1300
##  Median : 8110   Median :3.000   Median :  6.900   Median : 0.2800
##  Mean   : 8070   Mean   :3.029   Mean   :  7.076   Mean   : 0.3241
##  3rd Qu.:12106   3rd Qu.:4.000   3rd Qu.:  9.500   3rd Qu.: 0.6400
##  Max.   :16129   Max.   :8.000   Max.   : 34.400   Max.   : 3.6800
##
##    CitricAcid      ResidualSugar        Chlorides       FreeSulfurDioxide
##  Min.   :-3.2400   Min.   :-127.800   Min.   :-1.1710   Min.   :-555.00
##  1st Qu.: 0.0300   1st Qu.:  -2.000   1st Qu.:-0.0310   1st Qu.:   0.00
##  Median : 0.3100   Median :   3.900   Median : 0.0460   Median :  30.00
##  Mean   : 0.3084   Mean   :   5.419   Mean   : 0.0548   Mean   :  30.85
##  3rd Qu.: 0.5800   3rd Qu.:  15.900   3rd Qu.: 0.1530   3rd Qu.:  70.00
##  Max.   : 3.8600   Max.   : 141.150   Max.   : 1.3510   Max.   : 623.00
##                    NA's   :616        NA's   :638       NA's   :647
##  TotalSulfurDioxide    Density             pH          Sulphates
##  Min.   :-823.0     Min.   :0.8881   Min.   :0.480   Min.   :-3.1300
##  1st Qu.:  27.0     1st Qu.:0.9877   1st Qu.:2.960   1st Qu.: 0.2800
##  Median : 123.0     Median :0.9945   Median :3.200   Median : 0.5000
##  Mean   : 120.7     Mean   :0.9942   Mean   :3.208   Mean   : 0.5271
##  3rd Qu.: 208.0     3rd Qu.:1.0005   3rd Qu.:3.470   3rd Qu.: 0.8600
##  Max.   :1057.0     Max.   :1.0992   Max.   :6.130   Max.   : 4.2400
##  NA's   :682                         NA's   :395     NA's   :1210
##     Alcohol       LabelAppeal          AcidIndex          STARS
##  Min.   :-4.70   Min.   :-2.000000   Min.   : 4.000   Min.   :1.000
##  1st Qu.: 9.00   1st Qu.:-1.000000   1st Qu.: 7.000   1st Qu.:1.000
##  Median :10.40   Median : 0.000000   Median : 8.000   Median :2.000
##  Mean   :10.49   Mean   :-0.009066   Mean   : 7.773   Mean   :2.042
##  3rd Qu.:12.40   3rd Qu.: 1.000000   3rd Qu.: 8.000   3rd Qu.:3.000
##  Max.   :26.50   Max.   : 2.000000   Max.   :17.000   Max.   :4.000
##  NA's   :653                                          NA's   :3359
```

```r
dfc <- df
mylevels <- names(df[,2:15])
summary_plot <- df %>%
  gather() %>%
  mutate(facet = factor(key, levels=mylevels)) %>%
  ggplot(aes(value)) +
  facet_wrap(~ facet, scales = "free") +
  geom_histogram()  + theme_pander()  +
  theme(axis.text.y = element_text(size=7),
        strip.text.x = element_text(size= 9),
        axis.text.x = element_text(size=6),
        plot.title = element_text(hjust = 0.5, size=10),
        axis.title.y = element_text(size=8)) +
  labs(x=NULL, title="Wine Data Histograms")
summary_plot
```

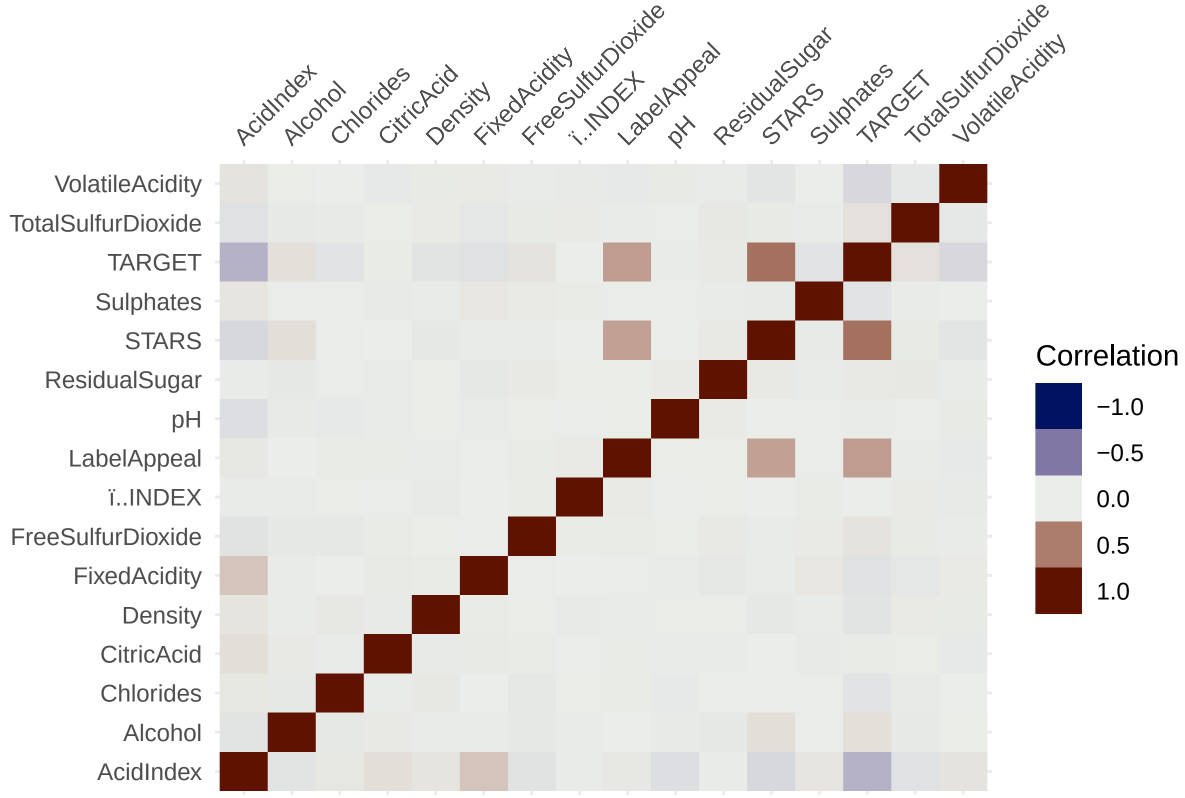


The distirubution of varialbes mostly on normal. Acid index is right-skewed. When we log it, its distribution appears normal. We maintain acid index as a logged variable. STARS appears in the dataset with a lot of NA entries. With the assumption that an unrated wine is overlooked and is likely to remain overlooked, we assume that rated wines would be more desirable. We imputed all NAs as 0 in our dataset. Below, we see that a simple linear model based on STARS as the independent variable can be improved when the STARS variable is augmented by 0s in place of NA

```
vis_miss(df)
```


i..INDEX (0%)
TARGET (0%)
FixedAcidity (0%)
VolatileAcidity (0%)
CitricAcid (0%)
ResidualSugar (4.81%)
Chlorides (4.99%)
FreeSulfurDioxide (5.06%)
TotalSulfurDioxide (5.33%)
Density (0%)
pH (3.09%)
Sulphates (9.46%)
Alcohol (5.1%)
LabelAppeal (0%)
AcidIndex (0%)
STARS (
0
5000
10000
Observations
Missing
(4%)
Present
(96%)

```r
vis_cor(df)
```



The correlations between many of our various variables are quite low for most of our variables. In our corrplot, only STARS, label index, alcohol and acid index have any visible correlation with our target. Acid index is negatively correlated, indicting that consumers don't like acidic wines

Data Preprocessing

Impute Missing Value

Imputing the missing value and applying some pre-processing steps to STARS, Acid Index and LabelAppeal variables

```
# Factors
df <- df %>%
  dplyr::select(-"ï..INDEX") %>%
  mutate(STARS = factor(STARS)) %>%
  mutate(STARS = fct_explicit_na(STARS,na_level = "0")) %>%
  mutate(LabelAppeal = factor(LabelAppeal)) %>%
  mutate(AcidIndex = if_else(AcidIndex <= 7,4L,AcidIndex)) %>%
  mutate(AcidIndex = if_else(AcidIndex == 8 | AcidIndex == 9 ,3L,AcidIndex)) %>%
  mutate(AcidIndex = if_else(AcidIndex == 10 | AcidIndex == 15 ,2L,AcidIndex)) %>%
  mutate(AcidIndex = if_else(AcidIndex == 16 | AcidIndex == 17 | AcidIndex == 11 | AcidIndex == 12 | Ac
  mutate(AcidIndex = factor(AcidIndex))
```

3. Data Preparation

Analysing different factors using the varibles for prediction models

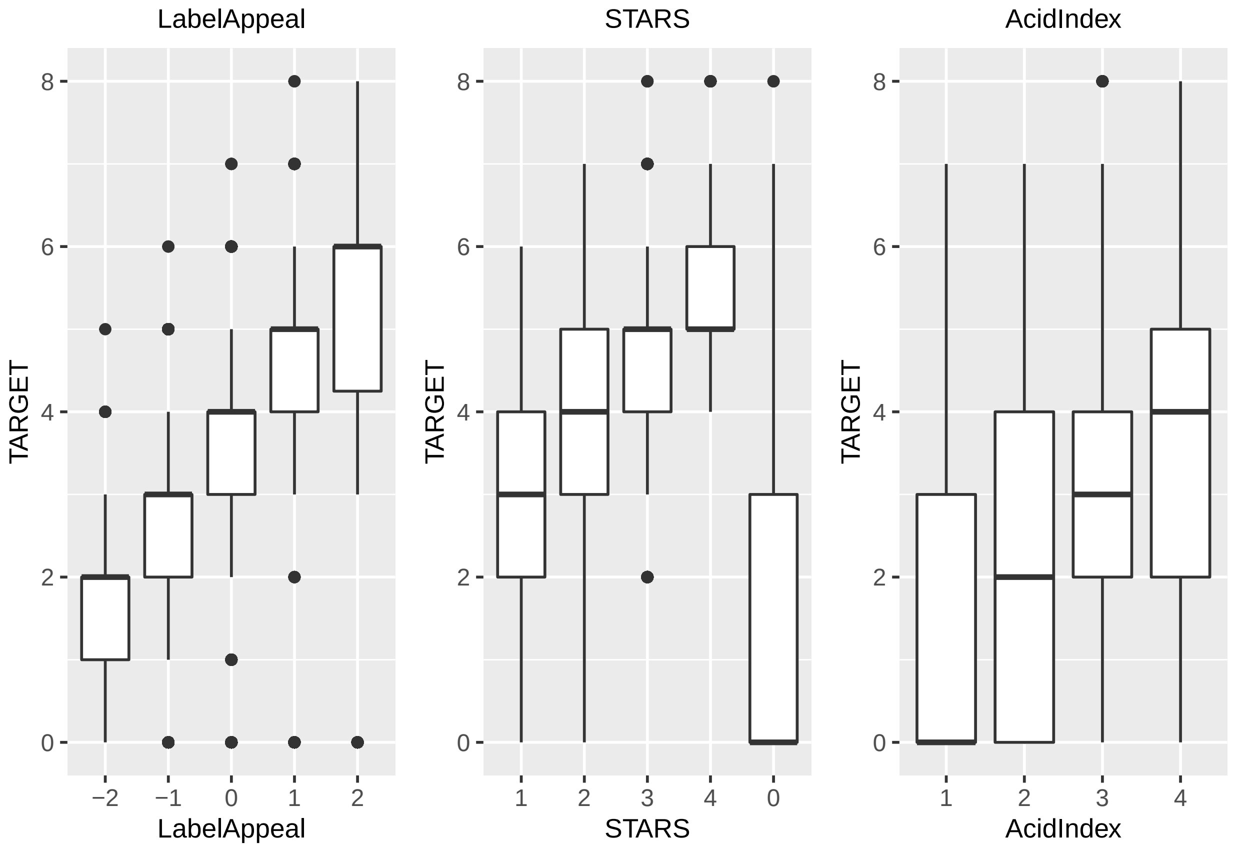
```r
tmp_data <- mice(df,maxit=3, method='pmm',seed=20, print=F)
df <- complete(tmp_data,1)

df$FixedAcidity <- abs(df$FixedAcidity)
df$VolatileAcidity <- abs(df$VolatileAcidity)
df$CitricAcid <- abs(df$CitricAcid)
df$ResidualSugar <- abs(df$ResidualSugar)
df$Chlorides <- abs(df$Chlorides)
df$FreeSulfurDioxide <- abs(df$FreeSulfurDioxide)
df$TotalSulfurDioxide <- abs(df$TotalSulfurDioxide)
df$Sulphates <- abs(df$Sulphates)
df$Alcohol <- abs(df$Alcohol)

str(df)
```

```
## '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 0.049 0.556 0.06 0.04 0.007 0.277 ...
##  $ FreeSulfurDioxide : num  4 15 214 22 167 37 287 523 213 62 ...
##  $ TotalSulfurDioxide: num  268 327 142 115 108 15 156 551 27 180 ...
##  $ 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 1.54 0.26 0.75 ...
##  $ Alcohol           : num  9.9 11.5 22 6.2 13.7 15.4 10.3 11.6 15 12.6 ...
##  $ LabelAppeal       : Factor w/ 5 levels "-2","-1","0",..: 3 2 2 2 3 3 3 4 3 3 ...
##  $ AcidIndex         : Factor w/ 4 levels "1","2","3","4": 3 4 3 4 3 1 3 4 4 3 ...
##  $ STARS             : Factor w/ 5 levels "1","2","3","4",..: 2 3 3 1 2 5 5 3 5 4 ...
```

```r
summary(df)
```

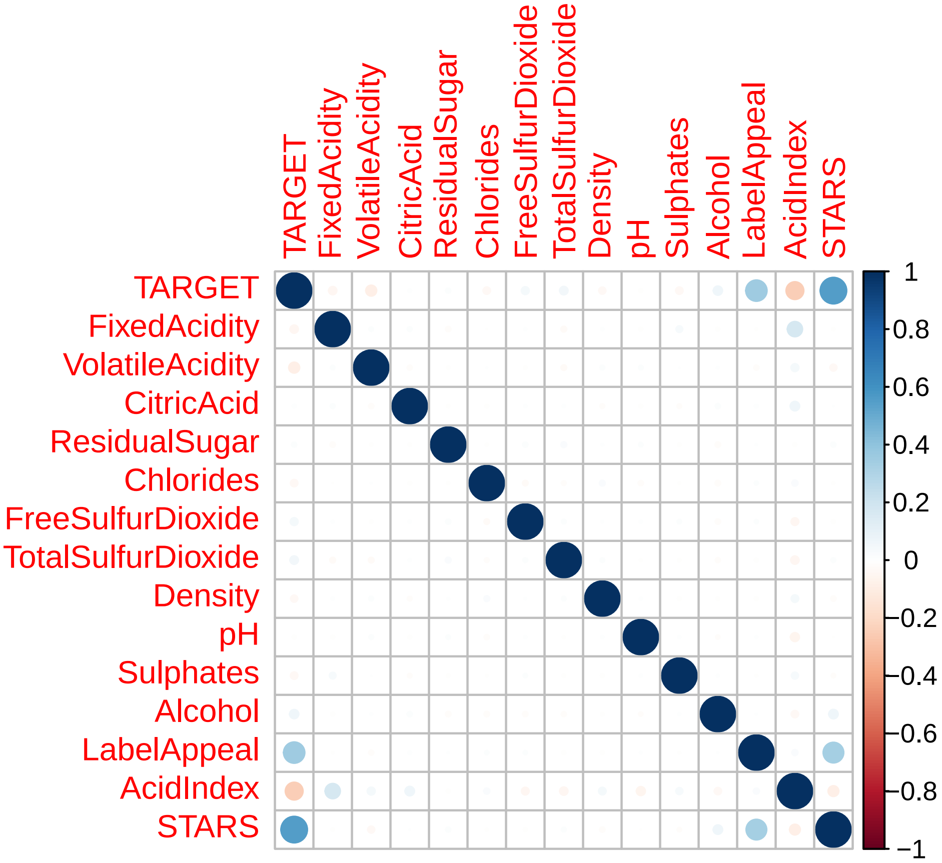
```
##      TARGET       FixedAcidity    VolatileAcidity    CitricAcid    
##  Min.   :0.000   Min.   : 0.000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:2.000   1st Qu.: 5.600   1st Qu.:0.2500   1st Qu.:0.2800  
##  Median :3.000   Median : 7.000   Median :0.4100   Median :0.4400  
##  Mean   :3.029   Mean   : 8.063   Mean   :0.6411   Mean   :0.6863  
##  3rd Qu.:4.000   3rd Qu.: 9.800   3rd Qu.:0.9100   3rd Qu.:0.9700  
##  Max.   :8.000   Max.   :34.400   Max.   :3.6800   Max.   :3.8600  
##  ResidualSugar      Chlorides      FreeSulfurDioxide TotalSulfurDioxide
##  Min.   :  0.00   Min.   :0.0000   Min.   :  0.0     Min.   :   0.0   
##  1st Qu.:  3.60   1st Qu.:0.0460   1st Qu.: 28.0     1st Qu.: 100.0   
##  Median : 13.00   Median :0.0980   Median : 56.0     Median : 154.0   
##  Mean   : 23.42   Mean   :0.2221   Mean   :106.6     Mean   : 204.9   
##  3rd Qu.: 38.80   3rd Qu.:0.3680   3rd Qu.:171.5     3rd Qu.: 264.0   
##  Max.   :141.15   Max.   :1.3510   Max.   :623.0     Max.   :1057.0   
##     Density             pH          Sulphates         Alcohol      LabelAppeal
##  Min.   :0.8881   Min.   :0.480   Min.   :0.000   Min.   : 0.00   -2: 504    
##  1st Qu.:0.9877   1st Qu.:2.960   1st Qu.:0.430   1st Qu.: 9.00   -1:3136    
##  Median :0.9945   Median :3.200   Median :0.590   Median :10.40   0 :5617    
##  Mean   :0.9942   Mean   :3.208   Mean   :0.846   Mean   :10.52   1 :3048    
##  3rd Qu.:1.0005   3rd Qu.:3.470   3rd Qu.:1.090   3rd Qu.:12.40   2 : 490
##  Max.   :1.0992   Max.   :6.130   Max.   :4.240   Max.   :26.50
##  AcidIndex STARS
##  1: 514    1:3042
##  2: 559    2:3570
##  3:5569    3:2212
##  4:6153    4: 612
##            0:3359
##
```

```
bp1 <- ggplot(df, aes(LabelAppeal,TARGET)) + geom_boxplot() +
  theme(axis.title = element_text(size=10),
        plot.title = element_text(hjust= 0.5, size = 10)) +
  labs(title = 'LabelAppeal')

bp2 <- ggplot(df, aes(STARS,TARGET)) + geom_boxplot() +
  theme(axis.title = element_text(size=10),
        plot.title = element_text(hjust= 0.5, size = 10)) +
  labs(title = 'STARS')

bp3 <- ggplot(df, aes(AcidIndex,TARGET)) + geom_boxplot() +
  theme(axis.title = element_text(size=10),
        plot.title = element_text(hjust= 0.5, size = 10)) +
  labs(title = 'AcidIndex')

grid.arrange(bp1, bp2, bp3, ncol = 3)
```



The corrplot shows lack of corelation, There does not seem to be any particularly strong correlation between variables.

```
train_data <- dfc[, -1]

corrplot(as.matrix(cor(train_data, use = "pairwise.complete")),method = "circle")
```



```
train_index <- createDataPartition(df$TARGET, p = .7, list = FALSE, times = 1)
train <- df[train_index,]
test <- df[-train_index,]
```

```
evaluate_model <- function(model, test_df, yhat = FALSE){
  temp <- data.frame(yhat=c(0:8), TARGET = c(0:8), n=c(0))

  if(yhat){
    test_df$yhat <- yhat
  } else {
    test_df$yhat <- round(predict.glm(model, newdata=test_df, type="response"), 0)
  }

  test_df <- test_df %>%
    group_by(yhat, TARGET) %>%
    tally() %>%
    mutate(accuracy = ifelse(yhat > TARGET, "Over", ifelse(yhat < TARGET, "Under", "Accurate"))) %>%
    mutate(cases_sold = ifelse(yhat > TARGET, TARGET, yhat) * n,
          glut = ifelse(yhat > TARGET, yhat - TARGET, 0) * n,
          missed_opportunity = ifelse(yhat < TARGET, TARGET - yhat, 0) * n) %>%
    mutate(net_cases_sold = cases_sold - glut,
          adj_net_cases_sold = cases_sold - glut - missed_opportunity)

  results <- test_df %>%
    group_by(accuracy) %>%
    summarise(n = sum(n)) %>%
    spread(accuracy, n)

  accurate <- results$Accurate
  over <- results$Over
  under <- results$Under

  cases_sold <- sum(test_df$cases_sold)
  net_cases_sold <- sum(test_df$net_cases_sold)
  adj_net_cases_sold <- sum(test_df$adj_net_cases_sold)
  missed_opportunity <- sum(test_df$missed_opportunity)
  glut <- sum(test_df$glut)

  confusion_matrix <- test_df %>%
    bind_rows(temp) %>%
    group_by(yhat, TARGET) %>%
    summarise(n = sum(n)) %>%
    spread(TARGET, n, fill = 0)

  return(list("confusion_matrix" = confusion_matrix, "results" = results, "df" = test_df, "accurate" = a
}
```

4. Build Models

The train and test data is splitted by 70/30 ration, the approach to modeling was to make strong use of the factor variable and limited use of the continuous variables given the uncertainty around the negative values. When continuous values were employed the absolute value of the variable is utilized in the model. We also employed three varieties of models in our analysis: Linear, Poisson, Negative Binomial Zero-Inflated.

A manually iterative process was employed to narrow the models down to the five contenders. Model summaries and confusion matrix data is presented for each model. The model evaluation section then picks a winner based upon a variety of factors, including: prediction ability (can the model predict all relevant value ranges), accuracy, AIC, BIC and LogLik.

MODEL 1 - POISSON 1

```
mod1 <- glm(TARGET ~ STARS + AcidIndex + LabelAppeal + Alcohol, family = poisson, train)
summary(mod1)
```

```
##
## Call:
## glm(formula = TARGET ~ STARS + AcidIndex + LabelAppeal + Alcohol,
##     family = poisson, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2403  -0.6497  -0.0100   0.4419   3.5922
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -0.058056   0.067077  -0.866   0.3868
## STARS2          0.329254   0.017097  19.258  < 2e-16 ***
## STARS3          0.452970   0.018553  24.414  < 2e-16 ***
## STARS4          0.573448   0.025693  22.320  < 2e-16 ***
## STARS0         -0.781756   0.023606 -33.116  < 2e-16 ***
## AcidIndex2      0.320547   0.059476   5.390 7.07e-08 ***
## AcidIndex3      0.570738   0.048398  11.793  < 2e-16 ***
## AcidIndex4      0.635386   0.048297  13.156  < 2e-16 ***
## LabelAppeal-1   0.259745   0.044948   5.779 7.52e-09 ***
## LabelAppeal0    0.435469   0.043843   9.933  < 2e-16 ***
## LabelAppeal1    0.562537   0.044593  12.615  < 2e-16 ***
## LabelAppeal2    0.705456   0.050303  14.024  < 2e-16 ***
## Alcohol         0.004194   0.001676   2.503   0.0123 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 16029.0  on 8957  degrees of freedom
## Residual deviance:  9506.9  on 8945  degrees of freedom
## AIC: 31883
##
## Number of Fisher Scoring iterations: 6
```

```r
pred <- predict(mod1, newdata=test, type='response')
predRound <- as.factor(round(pred,0)-1)
testData <- as.factor(test$TARGET)
levels(predRound) <- c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "0")
levels(testData) <- c("0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10")
cm <- confusionMatrix(predRound, testData)
cm$overall[1]
```

```
##  Accuracy
## 0.2142299
```

MODEL 2 - LINEAR 1

```r
mod2 <- lm(TARGET ~ STARS + AcidIndex + LabelAppeal + Alcohol, data = train)
summary(mod2)
```

```
##
## Call:
## lm(formula = TARGET ~ STARS + AcidIndex + LabelAppeal + Alcohol,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0097 -0.8497  0.0537  0.8263  5.6751
##
```

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)    0.636166   0.110001   5.783 7.57e-09 ***
## STARS2         1.062783   0.038789  27.399  < 2e-16 ***
## STARS3         1.641816   0.044708  36.723  < 2e-16 ***
## STARS4         2.328997   0.070851  32.872  < 2e-16 ***
## STARS0        -1.386655   0.039322 -35.264  < 2e-16 ***
## AcidIndex2     0.454331   0.094864   4.789 1.70e-06 ***
## AcidIndex3     1.017746   0.073200  13.904  < 2e-16 ***
## AcidIndex4     1.226141   0.073208  16.749  < 2e-16 ***
## LabelAppeal-1  0.429878   0.074910   5.739 9.86e-09 ***
## LabelAppeal0   0.864475   0.073032  11.837  < 2e-16 ***
## LabelAppeal1   1.304026   0.076194  17.115  < 2e-16 ***
## LabelAppeal2   1.943289   0.100985  19.243  < 2e-16 ***
## Alcohol        0.012843   0.003801   3.379  0.00073 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.306 on 8945 degrees of freedom
## Multiple R-squared:  0.5416, Adjusted R-squared:  0.541 
## F-statistic: 880.8 on 12 and 8945 DF,  p-value: < 2.2e-16
```

```
mod2_results <- evaluate_model(mod2, test)

pred <- predict(mod2, newdata=test)
predRound <- as.factor(round(pred,0))
levels(predRound) <- levels(as.factor(test$TARGET))
confusionMatrix(predRound, as.factor(test$TARGET))
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction   0   1   2   3   4   5   6   7   8
##          0   4   0   0   0   0   0   0   0   0
##          1  80  14  16   8   1   0   0   0   0
##          2 388  23  97 142  58  13   2   0   0
##          3 215  24 116 131  51  23   6   4   1
##          4 104   4  86 240 235  71  11   0   0
##          5  26   0  23 252 477 263  51   4   0
##          6   0   0   0  10 125 199 109  19   4
##          7   0   0   0   0   6  37  46  16   2
##          8   0   0   0   0   0   0   0   0   0
## 
## Overall Statistics
## 
##                Accuracy : 0.2265          
##                  95% CI : (0.2133, 0.2401)
##     No Information Rate : 0.2484          
##     P-Value [Acc > NIR] : 0.9993          
## 
##                   Kappa : 0.0912          
## 
##  Mcnemar's Test P-Value : NA              
## 
```

```
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity          0.004896 0.215385  0.28698  0.16731  0.24659  0.43399
## Specificity          1.000000 0.972163  0.82109  0.85593  0.82108  0.74219
## Pos Pred Value       1.000000 0.117647  0.13416  0.22942  0.31292  0.23996
## Neg Pred Value       0.787895 0.986283  0.92261  0.80037  0.76734  0.87486
## Prevalence           0.212927 0.016940  0.08809  0.20407  0.24837  0.15794
## Detection Rate       0.001042 0.003649  0.02528  0.03414  0.06125  0.06854
## Detection Prevalence 0.001042 0.031014  0.18843  0.14881  0.19573  0.28564
## Balanced Accuracy    0.502448 0.593774  0.55404  0.51162  0.53384  0.58809
##                      Class: 6 Class: 7 Class: 8
## Sensitivity           0.48444  0.37209 0.000000
## Specificity           0.90116  0.97601 1.000000
## Pos Pred Value        0.23391  0.14953      NaN
## Neg Pred Value        0.96559  0.99276 0.998176
## Prevalence            0.05864  0.01121 0.001824
## Detection Rate        0.02841  0.00417 0.000000
## Detection Prevalence  0.12145  0.02789 0.000000
## Balanced Accuracy     0.69280  0.67405 0.500000
```

MODEL 3 - POISSON 2

```
mod3 <- glm(TARGET ~ STARS + AcidIndex + LabelAppeal +  VolatileAcidity, family = poisson, train)
summary(mod3)
```

```
##
## Call:
## glm(formula = TARGET ~ STARS + AcidIndex + LabelAppeal + VolatileAcidity,
##     family = poisson, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2579  -0.6564  -0.0194   0.4495   3.5575
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.008339   0.065150   0.128   0.8981
## STARS2           0.328751   0.017101  19.224  < 2e-16 ***
## STARS3           0.454281   0.018535  24.510  < 2e-16 ***
## STARS4           0.576489   0.025648  22.477  < 2e-16 ***
## STARS0          -0.781269   0.023606 -33.096  < 2e-16 ***
## AcidIndex2       0.320105   0.059477   5.382 7.37e-08 ***
## AcidIndex3       0.569802   0.048399  11.773  < 2e-16 ***
## AcidIndex4       0.635259   0.048296  13.154  < 2e-16 ***
## LabelAppeal-1    0.256302   0.044943   5.703 1.18e-08 ***
## LabelAppeal0     0.431560   0.043836   9.845  < 2e-16 ***
## LabelAppeal1     0.557880   0.044579  12.514  < 2e-16 ***
## LabelAppeal2     0.701983   0.050307  13.954  < 2e-16 ***
## VolatileAcidity -0.028608   0.011192  -2.556   0.0106 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 16029.0  on 8957  degrees of freedom
## Residual deviance:  9506.6  on 8945  degrees of freedom
## AIC: 31883
##
## Number of Fisher Scoring iterations: 6
```

```
pred <- predict(mod3, newdata=test, type='response')
predRound <- as.factor(round(pred,0)-1)
testData <- as.factor(test$TARGET)
levels(predRound) <- c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "0")
levels(testData) <- c("0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10")
cm <- confusionMatrix(predRound, testData)
cm$overall[1]
```

```
##  Accuracy
## 0.2137086
```

MODEL 4 - LINEAR 2

```
mod4 <- lm(TARGET ~ STARS + AcidIndex + LabelAppeal +  VolatileAcidity, data = train)
summary(mod4)
```

```
##
## Call:
## lm(formula = TARGET ~ STARS + AcidIndex + LabelAppeal + VolatileAcidity,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0445 -0.8560  0.0402  0.8300  5.6142
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.84266    0.10372   8.125 5.09e-16 ***
## STARS2             1.06093    0.03880  27.346  < 2e-16 ***
## STARS3             1.64551    0.04467  36.841  < 2e-16 ***
## STARS4             2.33699    0.07077  33.022  < 2e-16 ***
## STARS0            -1.38570    0.03932 -35.242  < 2e-16 ***
## AcidIndex2         0.45400    0.09486   4.786 1.73e-06 ***
## AcidIndex3         1.01234    0.07321  13.828  < 2e-16 ***
## AcidIndex4         1.22311    0.07321  16.706  < 2e-16 ***
## LabelAppeal-1      0.41914    0.07488   5.597 2.24e-08 ***
## LabelAppeal0       0.85383    0.07300  11.696  < 2e-16 ***
## LabelAppeal1       1.29180    0.07614  16.965  < 2e-16 ***
## LabelAppeal2       1.93448    0.10097  19.159  < 2e-16 ***
## VolatileAcidity -0.09010    0.02496  -3.611 0.000307 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.306 on 8945 degrees of freedom
## Multiple R-squared:  0.5417, Adjusted R-squared:  0.5411
## F-statistic: 881.1 on 12 and 8945 DF,  p-value: < 2.2e-16
```

```r
pred <- predict(mod4, newdata=test)
predRound <- as.factor(round(pred,0))
levels(predRound) <- levels(as.factor(test$TARGET))
confusionMatrix(predRound, as.factor(test$TARGET))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1   2   3   4   5   6   7   8
##          0   4   0   0   0   0   0   0   0   0
##          1  81  15  16   8   1   0   0   0   0
##          2 380  22  96 127  50  10   1   0   0
##          3 224  24 119 149  62  26   7   4   1
##          4 102   4  73 222 229  70  11   0   0
##          5  26   0  34 261 471 266  54   4   0
##          6   0   0   0  16 134 197 112  19   4
##          7   0   0   0   0   6  37  40  16   2
##          8   0   0   0   0   0   0   0   0   0
##
## Overall Statistics
##
##                Accuracy : 0.2312
##                  95% CI : (0.2179, 0.2448)
##     No Information Rate : 0.2484
##     P-Value [Acc > NIR] : 0.9938
##
##                   Kappa : 0.0967
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 0 Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity          0.004896 0.230769  0.28402  0.19029  0.24029  0.43894
## Specificity          1.000000 0.971898  0.83138  0.84709  0.83287  0.73692
## Pos Pred Value       1.000000 0.123967  0.13994  0.24188  0.32208  0.23835
## Neg Pred Value       0.787895 0.986545  0.92320  0.80317  0.76839  0.87505
## Prevalence           0.212927 0.016940  0.08809  0.20407  0.24837  0.15794
## Detection Rate       0.001042 0.003909  0.02502  0.03883  0.05968  0.06932
## Detection Prevalence 0.001042 0.031535  0.17879  0.16054  0.18530  0.29085
## Balanced Accuracy    0.502448 0.601334  0.55770  0.51869  0.53658  0.58793
##                      Class: 6 Class: 7 Class: 8
## Sensitivity           0.49778  0.37209 0.000000
## Specificity           0.89756  0.97760 1.000000
## Pos Pred Value        0.23237  0.15842      NaN
## Neg Pred Value        0.96632  0.99277 0.998176
## Prevalence            0.05864  0.01121 0.001824
## Detection Rate        0.02919  0.00417 0.000000
## Detection Prevalence  0.12562  0.02632 0.000000
## Balanced Accuracy     0.69767  0.67484 0.500000
```

MODEL 5 - Zero-Inflated Negative Binomial (ZINB)

```r
mod5 <- zeroinfl(TARGET ~ STARS  + LabelAppeal + AcidIndex + TotalSulfurDioxide + VolatileAcidity, data

summary(mod5)
```

```
##
## Call:
## zeroinfl(formula = TARGET ~ STARS + LabelAppeal + AcidIndex + TotalSulfurDioxide +
##     VolatileAcidity, data = train, dist = "negbin")
##
## Pearson residuals:
##       Min        1Q    Median        3Q       Max
## -2.293046 -0.442584  0.006656  0.395610  4.994425
##
## Count model coefficients (negbin with log link):
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        4.464e-01  6.947e-02   6.427 1.31e-10 ***
## STARS2             1.270e-01  1.788e-02   7.107 1.19e-12 ***
## STARS3             2.285e-01  1.925e-02  11.870  < 2e-16 ***
## STARS4             3.292e-01  2.625e-02  12.538  < 2e-16 ***
## STARS0            -6.087e-02  2.558e-02  -2.380   0.0173 *
## LabelAppeal-1      4.646e-01  4.904e-02   9.475  < 2e-16 ***
## LabelAppeal0       7.405e-01  4.795e-02  15.443  < 2e-16 ***
## LabelAppeal1       9.278e-01  4.873e-02  19.038  < 2e-16 ***
## LabelAppeal2       1.091e+00  5.419e-02  20.126  < 2e-16 ***
## AcidIndex2        -6.354e-02  6.198e-02  -1.025   0.3053
## AcidIndex3         3.421e-02  5.018e-02   0.682   0.4954
## AcidIndex4         6.663e-02  5.000e-02   1.333   0.1827
## TotalSulfurDioxide -2.137e-05  3.663e-05  -0.583   0.5597
## VolatileAcidity   -1.395e-02  1.140e-02  -1.224   0.2208
## Log(theta)         1.720e+01  2.124e+00   8.097 5.62e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.722e+00  5.635e-01  -3.055 0.002248 **
## STARS2            -3.735e+00  3.887e-01  -9.609  < 2e-16 ***
## STARS3            -1.842e+01  4.657e+02  -0.040 0.968446
## STARS4            -1.862e+01  8.667e+02  -0.021 0.982863
## STARS0             2.111e+00  8.980e-02  23.512  < 2e-16 ***
## LabelAppeal-1      1.794e+00  5.331e-01   3.365 0.000765 ***
## LabelAppeal0       2.544e+00  5.308e-01   4.794 1.64e-06 ***
## LabelAppeal1       3.300e+00  5.352e-01   6.167 6.94e-10 ***
## LabelAppeal2       3.700e+00  5.873e-01   6.300 2.98e-10 ***
## AcidIndex2        -1.517e+00  2.527e-01  -6.002 1.94e-09 ***
## AcidIndex3        -2.241e+00  2.053e-01 -10.917  < 2e-16 ***
## AcidIndex4        -2.667e+00  2.076e-01 -12.852  < 2e-16 ***
## TotalSulfurDioxide -1.133e-03  2.569e-04  -4.412 1.02e-05 ***
## VolatileAcidity    1.002e-01  7.203e-02   1.391 0.164362
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 29568540.0922
## Number of iterations in BFGS optimization: 46
## Log-likelihood: -1.424e+04 on 29 Df
```

```
pred <- predict(mod5, newdata=test, type='response')
predRound <- as.factor(round(pred,0))
testData <- as.factor(test$TARGET)
cm <- confusionMatrix(predRound, testData)
cm$overall[1]
```

```
##  Accuracy
## 0.3388064
```

5. Select Model

Based on multiple model performance, the selection process is simple. Only the Zero-Inflated Negative Binomial model (ZINB) was able to meet or prediction ability criteria. Other models doesnt perform good to predict the zero values.

The ZINB model also outperformed all other models in terms of confusion matrix accuracy, AIC, BIC, logLik and length of model name. Summary results are set forth below.

```
# Select Models

mod1_result <- cbind(AIC=AIC(mod1), BIC = BIC(mod1), loglik=logLik(mod1))
mod2_result <- cbind(AIC=AIC(mod2),BIC = BIC(mod2), loglik=logLik(mod2))
mod3_result <- cbind(AIC=AIC(mod3),BIC = BIC(mod3), loglik=logLik(mod3))
mod4_result <- cbind(AIC=AIC(mod4), BIC = BIC(mod4), loglik=logLik(mod4))
mod5_result <- cbind(AIC=AIC(mod5), BIC = BIC(mod5), loglik=logLik(mod5))
model_comp <- rbind(mod1_result, mod2_result,mod3_result,mod4_result,mod5_result)


rownames(model_comp) <- c("mod1_result","mod2_result","mod3_result","mod4_result","mod5_result")

model_comp
```

```
##                  AIC      BIC    loglik
## mod1_result 31883.17 31975.48 -15928.59
## mod2_result 30214.48 30313.89 -15093.24
## mod3_result 31882.84 31975.15 -15928.42
## mod4_result 30212.87 30312.27 -15092.43
## mod5_result 28543.08 28748.99 -14242.54
```

6. Conclusion

ZINB (Zero-Inflated Negative Binomial model) Model Outperformed in missing value scenario when compared to Poisson and Linear Model