Wine Data Assignment

Cesar Espitia

CUNY SPS Data 621

Table of Contents

Abstract

This assignment focused on analyzing data about wines. The dataset contains over 12,000 records of commercially available wines. The data set has 15 variables, 1 outcome and 14 predictors, of different types such as continuous or factor type variables. The purpose for this assignment is to analyze the data, perform any data manipulation / clean-up and build two (2) poisson regressions, two (2) multiple negative binomial regression and two (2) linear regression models using only the data (or derivatives thereof) to predict if the number of cases of the wine were bought. The chosen model provided an AIC = 45499.4.

Keywords: wine, data621

Wine Data Assignment

The following is the analysis and write-up based upon my interpretation of the data and predict if an individual is likely to have an accident, and then if they do, what the claim amount may be.

Data Exploration

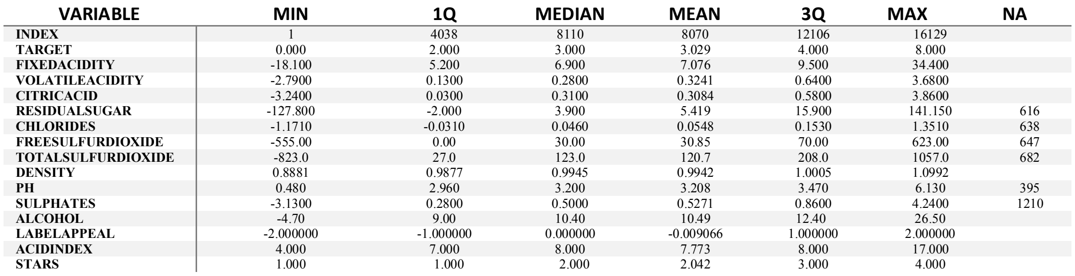
The purpose of this step is to get a 'feel' for the dataset. The following information describes the data from different angles including completeness, statistical summaries, visuals to determine the shape and effect of each variable and other items deemed pertinent.

Summary Statistics

The first step is to look at the data to determine some items including completeness and the shape of each variable.   The following are the results of summarizing the data in a table and the visualization of each variables density function (PDF).

Table 1

Summary Statistics for Moneyball Training Data



VARIABLE	MIN	1Q	MEDIAN	MEAN	3Q	MAX	NA
INDEX	1	4038	8110	8070	12106	16129	
TARGET	0.000	2.000	3.000	3.029	4.000	8.000	
FIXEDACIDITY	-18.100	5.200	6.900	7.076	9.500	34.400	
VOLATILEACIDITY	-2.7900	0.1300	0.2800	0.3241	0.6400	3.6800	
CITRICACID	-3.2400	0.0300	0.3100	0.3084	0.5800	3.8600	
RESIDUALSUGAR	-127.800	-2.000	3.900	5.419	15.900	141.150	616
CHLORIDES	-1.1710	-0.0310	0.0460	0.0548	0.1530	1.3510	638
FREESULFURDIOXIDE	-555.00	0.00	30.00	30.85	70.00	623.00	647
TOTALSULFURDIOXIDE	-823.0	27.0	123.0	120.7	208.0	1057.0	682
DENSITY	0.8881	0.9877	0.9945	0.9942	1.0005	1.0992	
PH	0.480	2.960	3.200	3.208	3.470	6.130	395
SULPHATES	-3.1300	0.2800	0.5000	0.5271	0.8600	4.2400	1210
ALCOHOL	-4.70	9.00	10.40	10.49	12.40	26.50	
LABELAPPEAL	-2.000000	-1.000000	0.000000	-0.009066	1.000000	2.000000	
ACIDINDEX	4.000	7.000	8.000	7.773	8.000	17.000	
STARS	1.000	1.000	2.000	2.042	3.000	4.000	

Note: Source: wine-training-data.csv

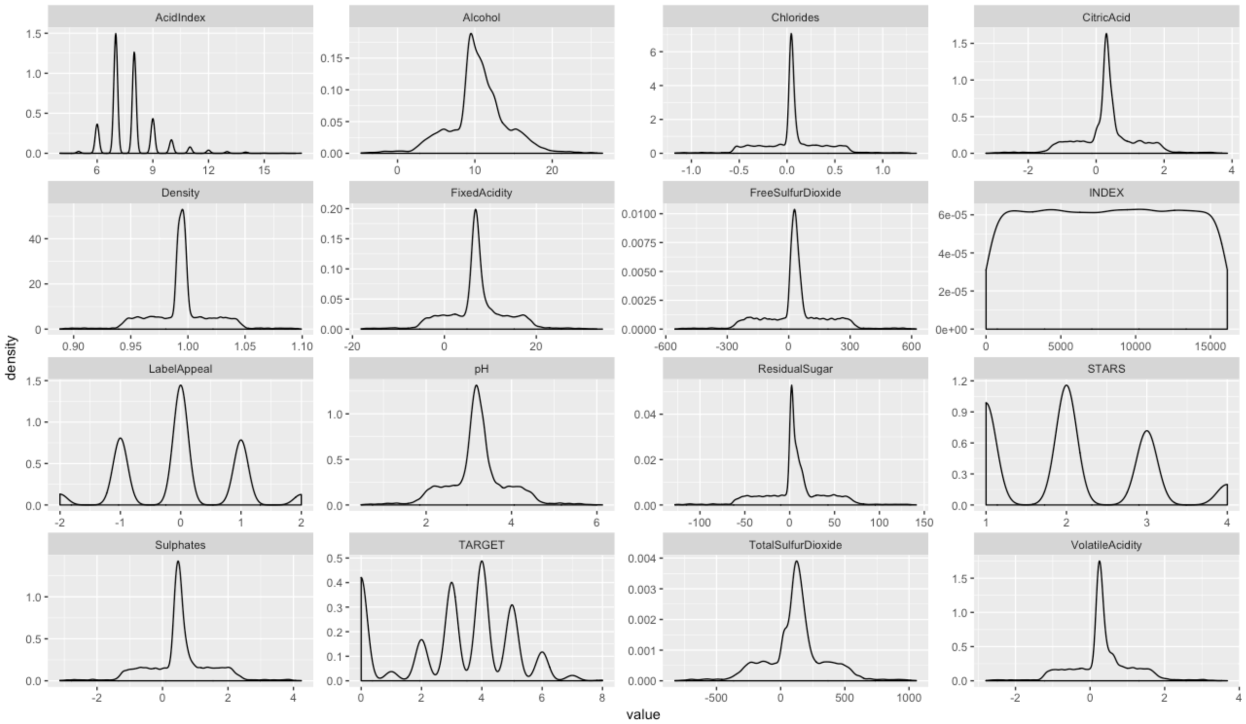
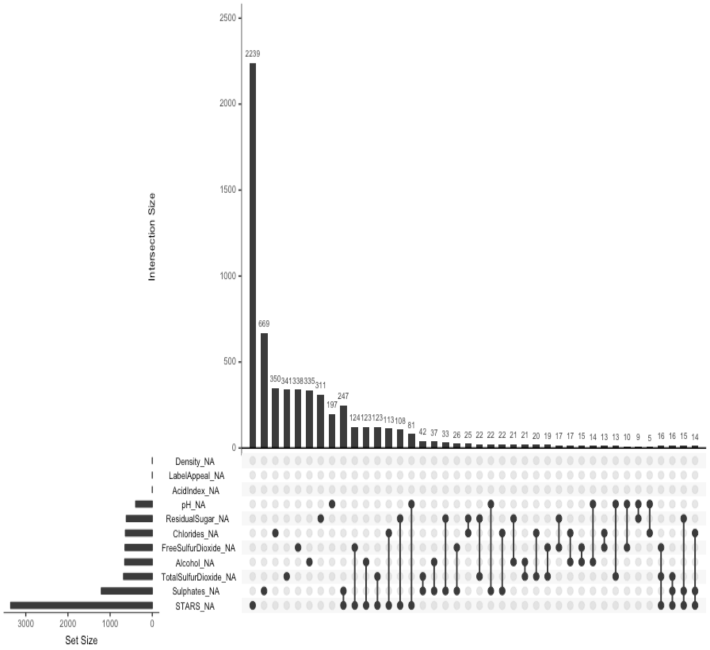
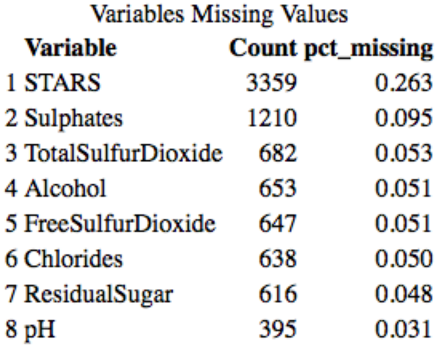


Figure 1. PDF for Each Dataframe Variable.

In looking at both, Table 1, Figure 1 and Appendix B (correlation matrix) together, we can note specific items that may skew our model building results.

NA: These incomplete cases will cause any correlation exercise to be incorrect or not possible. There are a few ways to deal with NAs including imputing the missing data or ignoring the variable altogether. For the purposes of this analysis, the variable ResidualSugar, Chlorides, FreeSulfurDioxide, TotalSulfurDioxide, pH, Sulphates, Alcohol and STARS have missing information. The highest offender is STARS with about 26.3% of the data missing while others are much lower than that.





Variables Missing Values

	Variable	Count	pct_missing
1	STARS	3359	0.263
2	Sulphates	1210	0.095
3	TotalSulfurDioxide	682	0.053
4	Alcohol	653	0.051
5	FreeSulfurDioxide	647	0.051
6	Chlorides	638	0.050
7	ResidualSugar	616	0.048
8	pH	395	0.031

Figure 2. NA Plot with Percentages

PDF: Figure 1 shows the PDF of each variable, this allows us to see if the data is normal or not. For all the variables, their shapes mimic the normal density functions that we are accustomed to except for AcidIndex and STARS. This is due to the fact that they are categorial values and even though they appear numeric they are not. All variables were left as is because the shape didn't warrant it.

Correlation: We look for correlated variables that we can make decisions on and determine which variable might be closely related to others either due to collinearity or other underlying factors that are visible at first glance in the dataset. Correlated variables bloat the model and don't produce any more insight than ignoring one of the two that show correlation. In our data, none of the variables show any particular correlation that would be cause for alarm and would require removal in order to avoid collinearity.

Data Preparation

The purpose of this step is to take the findings from the exploration and transform the data as needed. The following information describes the transformations done in order to prepare the data for model building and model selection.

NA: All missing values were imputed using the mean within each column even though it is not the most adequate for this data. The nearest neighbor method would have been more valid by using another variable to bin, but there was concern about causing bias due to calculating the mean on variables that have inherit bias in them (such as STARS which has the most). Therefore, the mean for the entire dataset (excluding NA values) was used for this analysis.

Absolute Transformation: For this dataset, nine (9) variables were transformed that were deemed overtly skewed in comparison to other variables in the dataset. FixedAcidity, VolatileAcidity, CitricAcid, ResidualSugar, Chloride, FreeSulfurDioxide, TotalSulfurDioxide, BoundSulfurDioxide, Sulphates and Alcohol were the variables transformed using the abs() function. ***CAVEAT: Please note that this was done as the presence of negative values do not make sense for any of the variables noted above. The assumption is that the data just had negative values in them that should have been positive. Other observations was that the data was Z-score transformed (as most are centered around 0). A log transform could have been done but shifting each variable by its min, would make it hard to translate the regression model coefficient results later on as they are not all being transformed in an equal manner.***

Variable Creation: For this dataset, two (2) new variables were created. BoundSulfurDioxide is the difference between Free and Total Sulfur Dioxide presnt in the wine and PerVol is the percentage of Volatile Acidity versus Total Acidity. Total Acidity was not generated as it should be accounted for in the ratio.

Correlation Check: Once these manipulations are done a correlation was done and due to the lack of correlation we do not need to remove any variables and can move forward.

Model Building for Outcome Variable TARGET_FLAG

The purpose of this step is to take the modified dataset and begin exploring potential models that will be used on the final dataset provided. The following information describes the six (6) models (2 poisson, 2 neg binomial and 2 linear regression) built for this step and the relevant analysis to provide reasons for model selection in the next step.

MODEL 1

The first model takes in the data as manipulated in step two. In this first model as a poisson, we have an AIC of 45561. The data in Table 2, shows that the model has an accuracy of 27.09%.

```
Call:
glm(formula = TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid +
   ResidualSugar + Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +
   BoundSulfurDioxide + Density + pH + Sulphates + Alcohol +
   as.factor(LabelAppeal) + as.factor(AcidIndex) + as.factor(STARS) +
   PerVol, family = poisson(), data = train)

Deviance Residuals:
   Min      1Q   Median     3Q     Max
-3.2127  -0.6516  -0.0030   0.4432   3.6940

Coefficients:
                          Estimate  Std. Error z value          Pr(>|z|)
(Intercept)                1.06253675  0.37156470   2.860          0.00424 **
FixedAcidity              -0.00072647  0.00126206  -0.576          0.56487
VolatileAcidity           -0.02903800  0.01123171  -2.585          0.00973 **
CitricAcid                 0.00869022  0.00834783   1.041          0.29787
ResidualSugar             -0.00001575  0.00020681  -0.076          0.93930
Chlorides                 -0.03234449  0.02218266  -1.458          0.14481
FreeSulfurDioxide          0.00006404  0.00005138   1.246          0.21263
TotalSulfurDioxide         0.00011812  0.00004809   2.456          0.01404 *
BoundSulfurDioxide        -0.00006537  0.00004433  -1.474          0.14035
Density                   -0.29557735  0.19193416  -1.540          0.12356
pH                        -0.00983182  0.00765360  -1.285          0.19893
Sulphates                 -0.01153271  0.00817562  -1.411          0.15836
Alcohol                    0.00461643  0.00144659   3.191          0.00142 **
as.factor(LabelAppeal)1    0.23924089  0.03800031   6.296
0.000000000306 ***
as.factor(LabelAppeal)2    0.42916835  0.03706591  11.579 <
0.0000000000000002 ***
as.factor(LabelAppeal)3    0.56226154  0.03771537  14.908 <
0.0000000000000002 ***
as.factor(LabelAppeal)4    0.69766946  0.04245421  16.433 <
0.0000000000000002 ***
as.factor(AcidIndex)5     -0.13380941  0.32271890  -0.415          0.67841
as.factor(AcidIndex)6     -0.10034777  0.31725980  -0.316          0.75178
as.factor(AcidIndex)7     -0.13264716  0.31700855  -0.418          0.67563
as.factor(AcidIndex)8     -0.16430607  0.31706657  -0.518          0.60431
as.factor(AcidIndex)9     -0.27397521  0.31739070  -0.863          0.38802
as.factor(AcidIndex)10    -0.43449994  0.31848283  -1.364          0.17248
as.factor(AcidIndex)11    -0.79602036  0.32208457  -2.471          0.01346 *
as.factor(AcidIndex)12    -0.80895430  0.32774169  -2.468          0.01358 *
as.factor(AcidIndex)13    -0.64343858  0.33066231  -1.946          0.05167 .
as.factor(AcidIndex)14    -0.74416112  0.34328561  -2.168          0.03018 *
as.factor(AcidIndex)15    -0.30132160  0.40394479  -0.746          0.45570
as.factor(AcidIndex)16    -0.95688354  0.54863387  -1.744          0.08114 .
as.factor(AcidIndex)17    -1.18518604  0.54861237  -2.160          0.03075 *
as.factor(STARS)2          0.31833077  0.01436884  22.154 <
0.0000000000000002 ***
as.factor(STARS)2.04175498092412 -0.75685033  0.01956973 -38.675 <
0.0000000000000002 ***
as.factor(STARS)3          0.43713915  0.01562442  27.978 <
0.0000000000000002 ***
as.factor(STARS)4          0.55871107  0.02166437  25.789 <
0.0000000000000002 ***
PerVol                    -0.05516995  0.05207826  -1.059          0.28943
---
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: 13549  on 12760  degrees of freedom                    Number of Fisher Scoring iterations: 6
AIC: 45561
```

Table 2. *Confusion Matrix Model 1*

True \ Pred	
Matched Cases Bought	3,466
Didn’t Match	12,795

Note that STARS, LabelAppeal and AcidIndex were taken as factors as they are categorial and not continuous in nature. No variables seem peculiar expect for AcidIndex, since it is an ascending scale the coefficients should be decrease in value as the index goes up which it doesn’t and is variable. As this is a correct interpretation of the variable, for now, this variable will be left in.

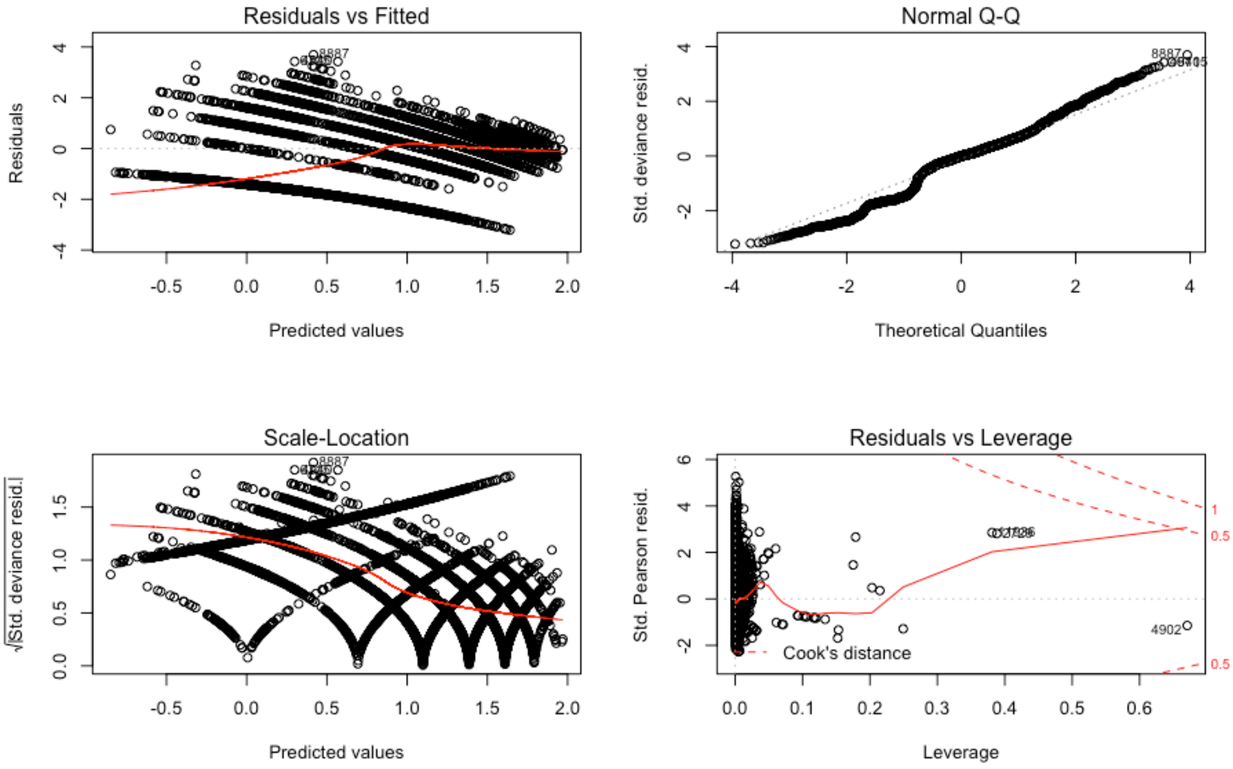


Figure 3. Model 1 (TARGET) Plots.

For this model, we can see that the Normal Q-Q plot shows unique character that is not only on the tails but also in the middle, this is due to the categorial nature of the TARGET flag. However, in looking at the residuals we see a condensed cluster on the left. The dispersion test also shows a p-value of 1 which means it is not the best model.

MODEL 2

The second model only takes into account the variables noted of significance from Model 1 (p-value < 0.05). In this second model as a poisson, we have an AIC of 45556. The data in Table 3, shows that the model has an accuracy of 27.03%.

```
glm(formula = TARGET ~ VolatileAcidity + TotalSulfurDioxide +
   Alcohol + as.factor(LabelAppeal) + as.factor(AcidIndex) +
   as.factor(STARS) + PerVol, family = poisson(), data = train)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-3.2471  -0.6496  -0.0005   0.4355   3.6907

Coefficients:
                         Estimate  Std. Error z value           Pr(>|z|)
(Intercept)               0.71353025  0.31931670   2.235          0.02545 *
VolatileAcidity          -0.03085150  0.01067228  -2.891          0.00384 **
TotalSulfurDioxide        0.00006467  0.00003195   2.024          0.04295 *
Alcohol                   0.00461529  0.00144657   3.191          0.00142 **
as.factor(LabelAppeal)1       0.23988496  0.03799700   6.313
0.000000000273 ***
as.factor(LabelAppeal)2       0.42949634  0.03706428  11.588 <
0.0000000000000002 ***
as.factor(LabelAppeal)3       0.56362465  0.03770892  14.947 <
0.0000000000000002 ***
as.factor(LabelAppeal)4       0.69761429  0.04244584  16.435 <
0.0000000000000002 ***
as.factor(AcidIndex)5      -0.12466124  0.32238208  -0.387          0.69899
as.factor(AcidIndex)6      -0.08925265  0.31691690  -0.282          0.77823
as.factor(AcidIndex)7      -0.12199358  0.31663296  -0.385          0.70003
as.factor(AcidIndex)8      -0.15350050  0.31666560  -0.485          0.62786
as.factor(AcidIndex)9      -0.26427415  0.31696999  -0.834          0.40442
as.factor(AcidIndex)10      -0.42663465  0.31805212  -1.341          0.17979
as.factor(AcidIndex)11      -0.79005656  0.32162571  -2.456          0.01403 *
as.factor(AcidIndex)12      -0.80327975  0.32728632  -2.454          0.01411 *
as.factor(AcidIndex)13      -0.63916256  0.33019908  -1.936          0.05291 .
as.factor(AcidIndex)14      -0.73826506  0.34274553  -2.154          0.03124 *
as.factor(AcidIndex)15      -0.28283782  0.40345858  -0.701          0.48328
as.factor(AcidIndex)16      -0.95458004  0.54800017  -1.742          0.08152 .
as.factor(AcidIndex)17      -1.19689236  0.54811293  -2.184          0.02899 *
as.factor(STARS)2          0.31814639  0.01436122  22.153 <
0.0000000000000002 ***
as.factor(STARS)2.04175498092412 -0.75871740  0.01956057 -38.788 <
0.0000000000000002 ***
as.factor(STARS)3          0.43756789  0.01561931  28.015 <
0.0000000000000002 ***
as.factor(STARS)4          0.55870679  0.02166337  25.790 <
0.0000000000000002 ***
PerVol                   -0.04074099  0.04313558  -0.944          0.34492
---
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: 13562  on 12769  degrees of freedom
AIC: 45556

Number of Fisher Scoring iterations: 6
```

Table 3. *Confusion Matrix Model 2*

True \ Pred	
Matched Cases Bought	3,458
Didn’t Match	12,795

Note that STARS, LabelAppeal and AcidIndex were taken as factors as they are categorial and not continuous in nature. No variables seem peculiar expect for AcidIndex, since it is an ascending scale the coefficients should be decrease in value as the index goes up which it doesn't and is variable. As this is a correct interpretation of the variable, for now, this variable will be left in.

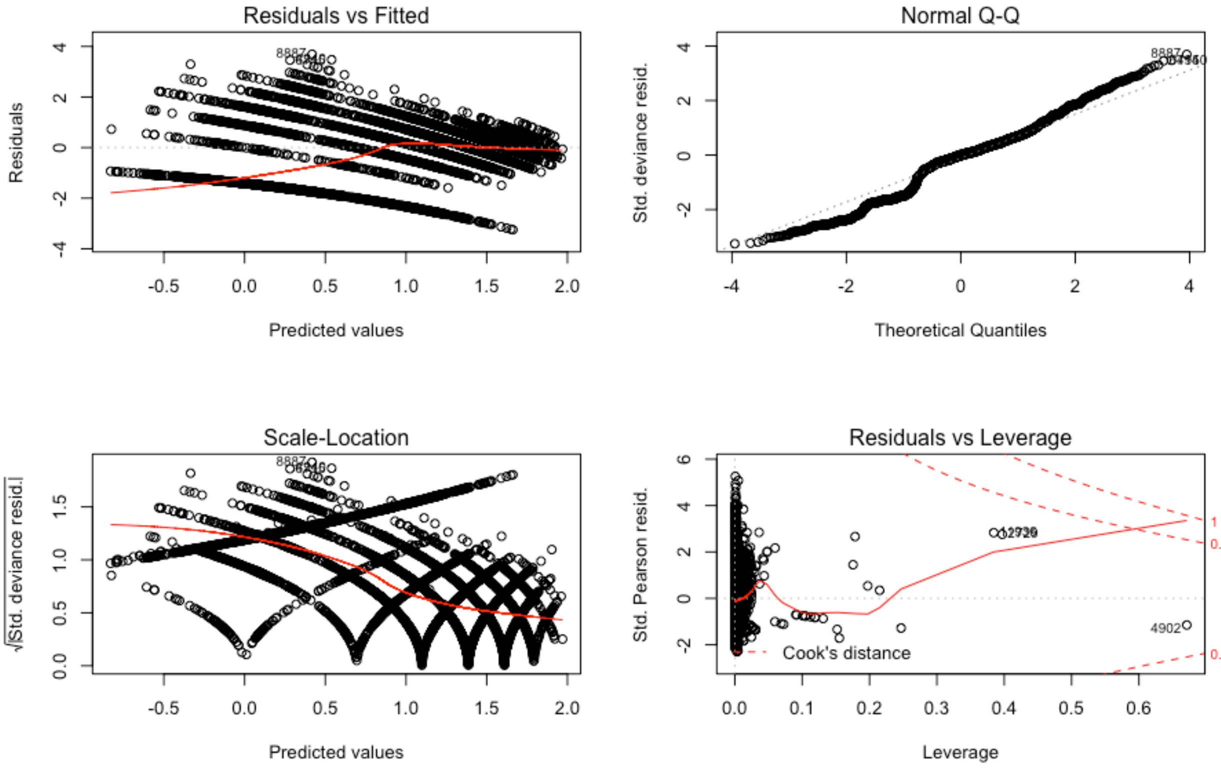


Figure 4. Model 2 (TARGET) Plots.

For this model, we can see that the Normal Q-Q plot shows unique character that is not only on the tails but also in the middle, this is due to the categorial nature of the TARGET flag. However, in looking at the residuals we see a condensed cluster on the left. The dispersion test also shows a p-value of 1 which means it is not the best model.

MODEL 3

The third model is a negative binomial model which is meant to fit categorical count data in a more effective manner. These models were built using the MASS package. In this third model, we have an AIC of 45564. The data in Table 3, shows that the model has an accuracy of 27.09%.

```
glm.nb(formula = TARGET ~ FixedAcidity + VolatileAcidity + CitricAcid +
   ResidualSugar + Chlorides + FreeSulfurDioxide + TotalSulfurDioxide +
   BoundSulfurDioxide + Density + pH + Sulphates + Alcohol +
   as.factor(LabelAppeal) + as.factor(AcidIndex) + as.factor(STARS) +
   PerVol, data = train, init.theta = 40922.4051, link = log)

Deviance Residuals:
   Min       1Q   Median       3Q      Max
-3.2126  -0.6516  -0.0030   0.4431   3.6939

Coefficients:
                          Estimate  Std. Error z value            Pr(>|z|)
(Intercept)                1.06256763  0.37158580   2.860            0.00424 **
FixedAcidity              -0.00072650  0.00126212  -0.576            0.56487
VolatileAcidity           -0.02903909  0.01123219  -2.585            0.00973 **
CitricAcid                 0.00869043  0.00834821   1.041            0.29788
ResidualSugar             -0.00001575  0.00020682  -0.076            0.93929
Chlorides                 -0.03234520  0.02218366  -1.458            0.14482
FreeSulfurDioxide          0.00006404  0.00005138   1.246            0.21263
TotalSulfurDioxide         0.00011812  0.00004809   2.456            0.01404 *
BoundSulfurDioxide        -0.00006537  0.00004434  -1.474            0.14035
Density                   -0.29558179  0.19194284  -1.540            0.12357
pH                        -0.00983270  0.00765394  -1.285            0.19891
Sulphates                 -0.01153325  0.00817598  -1.411            0.15836
Alcohol                    0.00461635  0.00144666   3.191            0.00142 **
as.factor(LabelAppeal)1    0.23924036  0.03800119   6.296      0.000000000306 ***
as.factor(LabelAppeal)2    0.42916666  0.03706677  11.578 < 0.0000000000000002
***
as.factor(LabelAppeal)3    0.56225795  0.03771630  14.908 < 0.0000000000000002
***
as.factor(LabelAppeal)4    0.69766527  0.04245561  16.433 < 0.0000000000000002
***
as.factor(AcidIndex)5     -0.13382982  0.32273859  -0.415            0.67838
as.factor(AcidIndex)6     -0.10036698  0.31727930  -0.316            0.75175
as.factor(AcidIndex)7     -0.13266700  0.31702803  -0.418            0.67560
as.factor(AcidIndex)8     -0.16432660  0.31708605  -0.518            0.60429
as.factor(AcidIndex)9     -0.27399928  0.31741019  -0.863            0.38801
as.factor(AcidIndex)10    -0.43452627  0.31850230  -1.364            0.17248
as.factor(AcidIndex)11    -0.79605119  0.32210392  -2.471            0.01346 *
as.factor(AcidIndex)12    -0.80898684  0.32776092  -2.468            0.01358 *
as.factor(AcidIndex)13    -0.64346919  0.33068163  -1.946            0.05167 .
as.factor(AcidIndex)14    -0.74418859  0.34330458  -2.168            0.03018 *
as.factor(AcidIndex)15    -0.30134808  0.40396482  -0.746            0.45568
as.factor(AcidIndex)16    -0.95692082  0.54865195  -1.744            0.08114 .
as.factor(AcidIndex)17    -1.18522561  0.54862910  -2.160            0.03075 *
as.factor(STARS)2          0.31833116  0.01436939  22.153 < 0.0000000000000002
***
as.factor(STARS)2.04175498092412 -0.75684944  0.01957014 -38.674 <
0.0000000000000002 ***
as.factor(STARS)3          0.43714031  0.01562508  27.977 < 0.0000000000000002
***
as.factor(STARS)4          0.55871330  0.02166558  25.788 < 0.0000000000000002
***
PerVol                    -0.05517181  0.05208054  -1.059            0.28944
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(40922.41) family taken to be 1)

    Null deviance: 22860  on 12794  degrees of freedom
Residual deviance: 13549  on 12760  degrees of freedom
AIC: 45564

Number of Fisher Scoring iterations: 1


              Theta:  40922
          Std. Err.:  34326
Warning while fitting theta: iteration limit reached

 2 x log-likelihood:  -45491.65
```

Table 4. *Confusion Matrix Model 3*

True \ Pred	
Matched Cases Bought	3,466
Didn’t Match	12,795

This model doesn’t do better than Model 1 which was a poisson and has a similar accuracy rate. While note as simple it is better than flipping a coin, the accuracy leaves much to be desired.

What is curious is at the higher counts, the matrix in the upper diagonal quadrant is less likely to be predicted vs the lower diagonal.

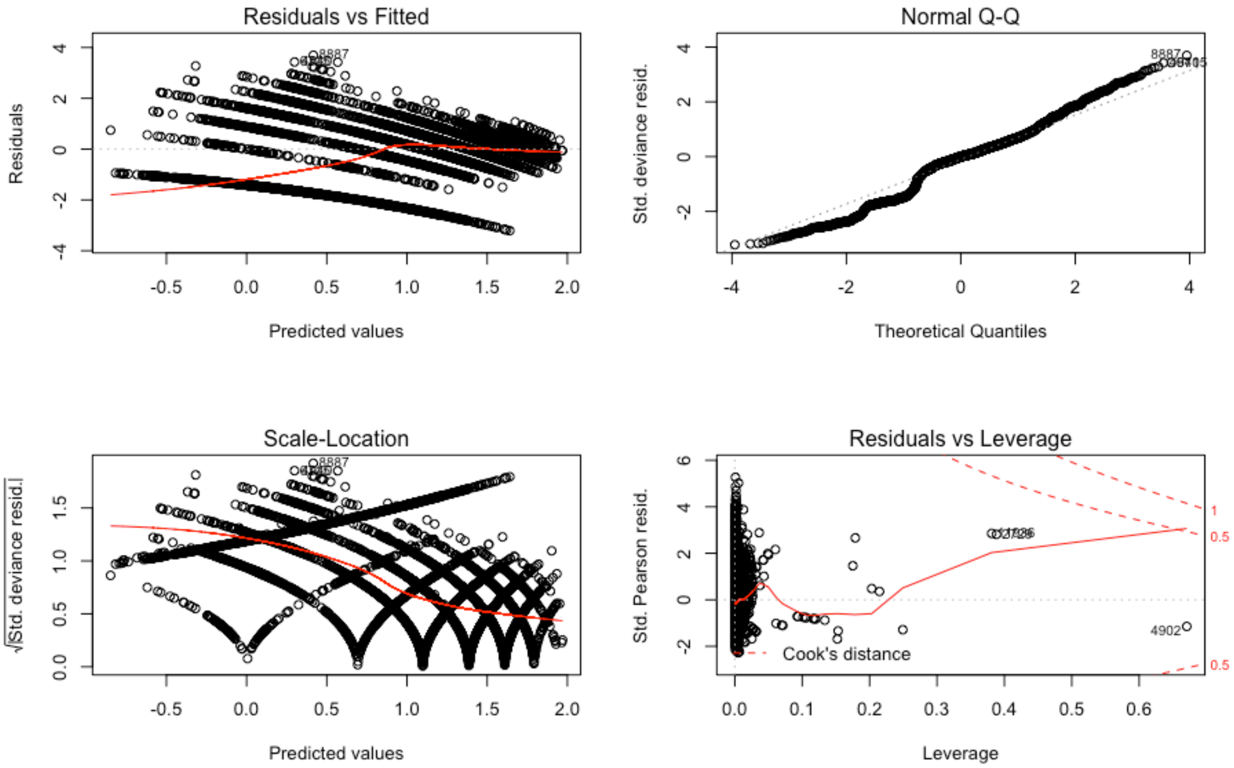


Figure 5. Model 3. (TARGET) Plots.

For this model, we can see that the Normal Q-Q plot is completely different than the others. The lower tail deviates more than in Model 1 and 2. The residuals also clump less along the left.

MODEL 4

The fourth model is another negative binomial model which is meant to fit categorical count data in a more effective manner. These models were built using the MASS package and only include those of significance that were found in Model 3. The same variables were taken as factors for this model also as in Model 1, 2 and 3. In this fourth model, we have an AIC of 45558. The data in Table 3, shows that the model has an accuracy of 27.03%.

```
glm.nb(formula = TARGET ~ VolatileAcidity + TotalSulfurDioxide +
    Alcohol + as.factor(LabelAppeal) + as.factor(AcidIndex) +
    as.factor(STARS) + PerVol, data = train, init.theta = 40886.26992,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.2470  -0.6496  -0.0005   0.4354   3.6906

Coefficients:
                          Estimate  Std. Error z value            Pr(>|z|)
(Intercept)             0.71355226  0.31933620   2.234             0.02545 *
VolatileAcidity        -0.03085266  0.01067275  -2.891             0.00384 **
TotalSulfurDioxide      0.00006468  0.00003195   2.024             0.04294 *
Alcohol                 0.00461521  0.00144663   3.190             0.00142 **
as.factor(LabelAppeal)1 0.23988448  0.03799789   6.313      0.000000000273 ***
as.factor(LabelAppeal)2 0.42949469  0.03706514  11.588 < 0.0000000000000002 ***
as.factor(LabelAppeal)3 0.56362112  0.03770984  14.946 < 0.0000000000000002 ***
as.factor(LabelAppeal)4 0.69761012  0.04244725  16.435 < 0.0000000000000002 ***
as.factor(AcidIndex)5  -0.12468053  0.32240182  -0.387             0.69896
as.factor(AcidIndex)6  -0.08927068  0.31693642  -0.282             0.77820
as.factor(AcidIndex)7  -0.12201221  0.31665248  -0.385             0.70000
as.factor(AcidIndex)8  -0.15351977  0.31668512  -0.485             0.62784
as.factor(AcidIndex)9  -0.26429701  0.31698951  -0.834             0.40441
as.factor(AcidIndex)10 -0.42665984  0.31807163  -1.341             0.17979
as.factor(AcidIndex)11 -0.79008634  0.32164510  -2.456             0.01403 *
as.factor(AcidIndex)12 -0.80331117  0.32730558  -2.454             0.01412 *
as.factor(AcidIndex)13 -0.63919219  0.33021843  -1.936             0.05291 .
as.factor(AcidIndex)14 -0.73829156  0.34276453  -2.154             0.03125 *
as.factor(AcidIndex)15 -0.28286247  0.40347863  -0.701             0.48327
as.factor(AcidIndex)16 -0.95461650  0.54801814  -1.742             0.08152 .
as.factor(AcidIndex)17 -1.19693172  0.54812968  -2.184             0.02899 *
as.factor(STARS)2       0.31814672  0.01436176  22.152 < 0.0000000000000002 ***
as.factor(STARS)2.04175498092412 -0.75871659  0.01956098 -38.787 < 0.0000000000000002 ***
as.factor(STARS)3       0.43756899  0.01561998  28.013 < 0.0000000000000002 ***
as.factor(STARS)4       0.55870896  0.02166458  25.789 < 0.0000000000000002 ***
PerVol                 -0.04074236  0.04313746  -0.944             0.34493
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(40886.27) family taken to be 1)

    Null deviance: 22860  on 12794  degrees of freedom
Residual deviance: 13561  on 12769  degrees of freedom
AIC: 45558

Number of Fisher Scoring iterations: 1


              Theta:  40886
          Std. Err.:  34285
Warning while fitting theta: iteration limit reached

 2 x log-likelihood:  -45504.06
```

Table 5. *Confusion Matrix Model 4*

True \ Pred	
Matched Cases Bought	3,458
Didn't Match	12,795

This model doesn't do better than Model 3 which was also a negative binomial and has a similar accuracy rate. While note as simple it is better than flipping a coin, the accuracy leaves much to be desired. What is curious is at the higher counts, the matrix in the upper diagonal quadrant is less likely to be predicted vs the lower diagonal.

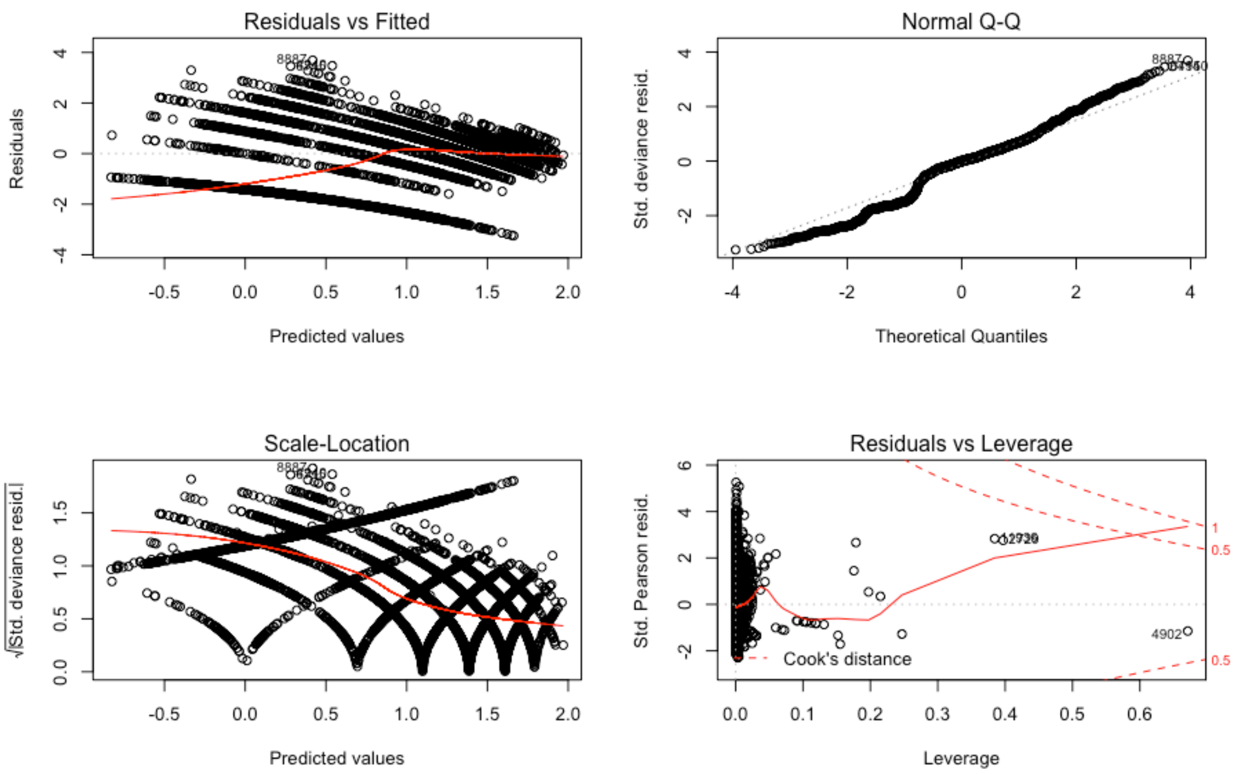


Figure 6. Model 4. (TARGET) Plots.

For this model, we can see that the Normal Q-Q plot is completely different than the others. The lower tail deviates more than in Model 1 and 2. The residuals also clump less along the left.

MODEL 5

The fifth model takes in the data as manipulated in step two (with variables imputed and removed). In this first model, we have an $R^2 = 0.2779$ and p-value < 0.05. The data in Figure 3, shows that there is not heteroscedastic and has a positive trend on the predicted vs fitted values.

```
lm(formula = TARGET ~ ., data = train)

Residuals:
   Min      1Q  Median      3Q     Max
```

```
-5.0189 -0.7380  0.3737  1.1294  4.6454

Coefficients:
                Estimate Std. Error t value          Pr(>|t|)
```

```
(Intercept)        4.3757840  0.5573755   7.851  0.0000000000000447 ***
FixedAcidity      -0.0036799  0.0035701  -1.031            0.302683
VolatileAcidity   -0.1559722  0.0312290  -4.994  0.00000059778214213 ***
CitricAcid         0.0551764  0.0239298   2.306            0.021140 *
ResidualSugar     -0.0001641  0.0005880  -0.279            0.780230
Chlorides         -0.1415407  0.0626819  -2.258            0.023957 *
FreeSulfurDioxide  0.0004751  0.0001478   3.214            0.001312 **
TotalSulfurDioxide 0.0007186  0.0001376   5.222  0.00000017982823371 ***
Density           -1.3781218  0.5464768  -2.522            0.011687 *
pH                -0.0633858  0.0216939  -2.922            0.003486 **
Sulphates         -0.0665696  0.0229855  -2.896            0.003784 **
Alcohol            0.0210964  0.0041090   5.134  0.00000028748895271 ***
```

```
LabelAppeal         0.6034374  0.0169723  35.554 < 0.0000000000000002 ***
AcidIndex          -0.3300359  0.0112552 -29.323 < 0.0000000000000002 ***
STARS               0.7178055  0.0195731  36.673 < 0.0000000000000002 ***
BoundSulfurDioxide -0.0004583  0.0001280  -3.581            0.000343 ***
PerVol             -0.1285625  0.1472273  -0.873            0.382557
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.638 on 12778 degrees of freedom
Multiple R-squared:  0.2779,                    Adjusted R-squared:  0.277
F-statistic: 307.3 on 16 and 12778 DF,  p-value: < 0.00000000000000022
```

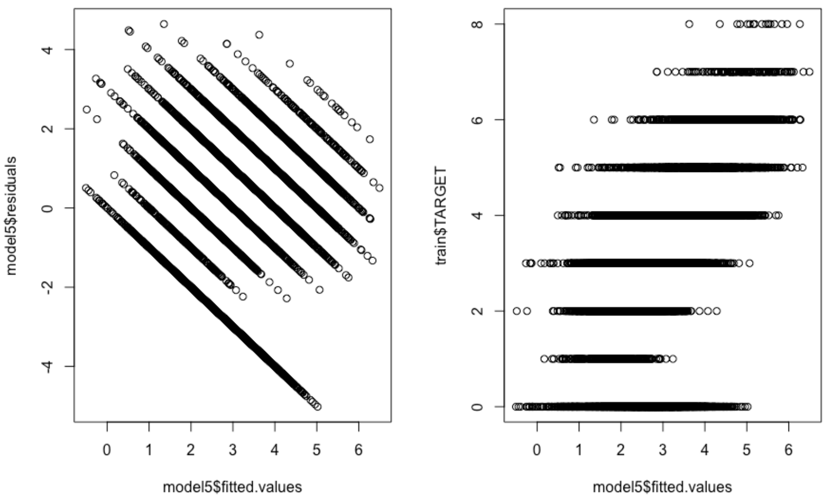


Figure 7. Model Check for Residual Shape and Model vs. Actuals

What is peculiar in the results is that all variables except for FixedAcidity, ResidualSugar and PerVol were found to be insignificant. This means that all variables are being used which is a sign of overfitting.

MODEL 6

The sixth model only takes into account the variables noted of significance from Model 5 (p-value < 0.05). In this second model, we have an $R^2 = 0.2771$ and p-value < 0.05 which is a worsening in the model capability.

```
lm(formula = TARGET ~ ., data = trainmod2)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-5.0101 -0.7355  0.3733  1.1267  4.6520 

Coefficients:
                     Estimate Std. Error t value             Pr(>|t|)    
(Intercept)         4.3497981  0.5567548   7.813  0.00000000000000603 ***
VolatileAcidity    -0.1710061  0.0261125  -6.549  0.00000000006021103 ***
CitricAcid          0.0554216  0.0239248   2.316             0.020547 *  
Chlorides          -0.1420447  0.0626707  -2.267             0.023436 *  
FreeSulfurDioxide   0.0004756  0.0001478   3.218             0.001294 ** 
TotalSulfurDioxide  0.0007177  0.0001376   5.216  0.00000018534856613 ***
Density            -1.3735422  0.5464180  -2.514             0.011959 *  
pH                 -0.0638339  0.0216882  -2.943             0.003254 ** 
Sulphates          -0.0670812  0.0229781  -2.919             0.003514 ** 
Alcohol             0.0210671  0.0041083   5.128  0.00000029718809899 ***
LabelAppeal         0.6036101  0.0169705  35.568 < 0.0000000000000002 ***
AcidIndex          -0.3319226  0.0110520 -30.033 < 0.0000000000000002 ***
STARS               0.7178463  0.0195712  36.679 < 0.0000000000000002 ***
BoundSulfurDioxide -0.0004568  0.0001279  -3.570             0.000358 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.638 on 12781 degrees of freedom
Multiple R-squared:  0.2778,    Adjusted R-squared:  0.2771 
F-statistic: 378.2 on 13 and 12781 DF,  p-value: < 0.00000000000000022
```

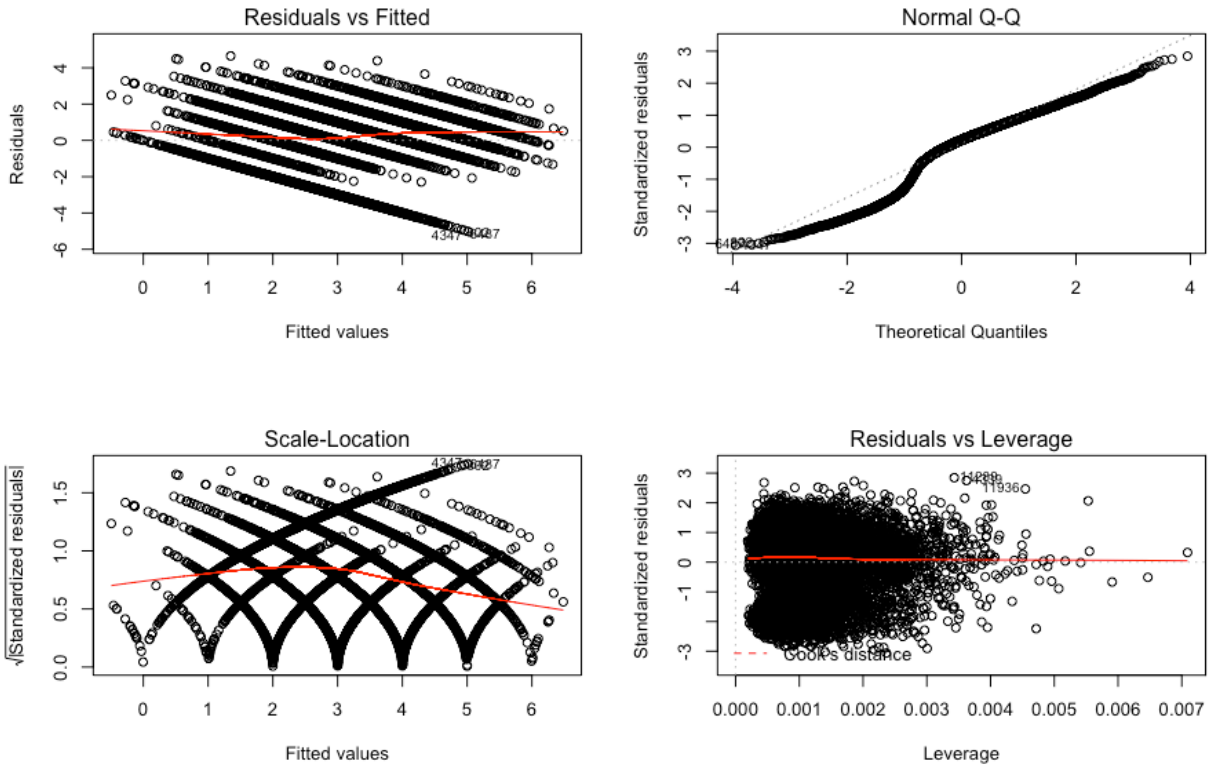


Figure 8. Model 6 Plots (Residuals vs Fitted and QQ)

This model does worse than any of the others as the tails on the lower left are deviating from the normal line diagonally.

METHODOLOGY

Familiarity with the dataset subject is low and the numerous issues with the data and its meaning were hard to ignore and therefore the methodology will be more closely related to the statistical information presented. In this case, only one factor, AIC for Models 1-4 and R^2 for Models 5 and 6 will be used for this analysis.

Table 6. *Model Criteria Selection*

Criteria	Model 1 Poisson	**Model 2 Poisson**	Model 3 Neg Bin	Model 4 Neg Bin	Model 5 Linear	Model 6 Linear
AIC	45,561	45,556	45,564	45,558		
R^2					0.2779%	0.2771%

The capabilities of any of these models to predict well is nuanced and minimal. We are better off flipping a coin, however based upon the data, we will chose Model 2 as it is the best AIC with the least amount of variables.

TEST DATA

The dataset had 3,335 entries and 15 columns and was modified to fit the final variables and scaling used in Model 1 from above. This means that the same process of adjustments and abs transformations was done in order to be able to use the model correctly. The final predicted values are based upon a normalized value from the test data. The data is shown as follows with the corresponding summaries for the spread of the data.

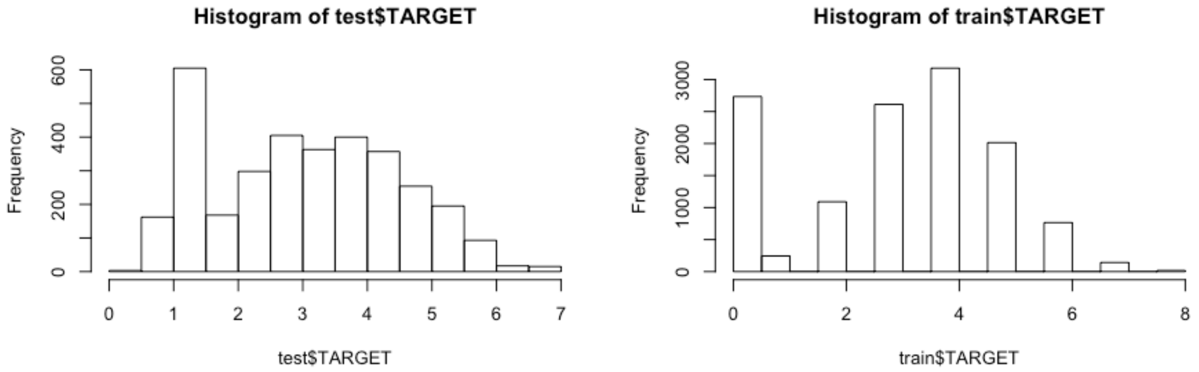


Figure 9. TARGET Histograms (test$TARGET on Model 2 vs train$TARGET).

Table 7. *Predicted Statistics vs Summary of Model 1 Predicted Values for TARGET_FLAG*

Dataset	0	1	2	3	4	5	6	7	8
Test	4.9%	23.2%	21.1%	22.9%	18.3%	8.6%	1.0%	0.0%	0.0%
Train	21.4%	1.9%	8.5%	20.4%	24.8%	15.7%	6.0%	1.1%	0.1%

Table 6 above is only meant as a comparison but it does highlight that the test data had predictions in the lower case counts than the double binomial histogram of the training data). This has to do with the nature of the predictions being continuous and then being floor to fit the integer counts. Overall, the analysis of the data is being influenced by the transformation of the data set as negative values were transformed to positives without clear understanding of the data entry.

Conclusion

Six (6) models were presented after exploring and manipulating the data as necessary. With using a singular criteria approach for this exercise, it became clear that the Model 2 was selected and provided an AIC of 45,556 for TARGET. If more time were available, the clean-up of the negative variables would be explored to create more factored variables instead of continuous variables that were presented and could have provided better insight into the data set.

Appendix A: R Code

```
---
title: "Data 621"
author: 'Cesar Espitia HW #5
date: "7/19/2018"
output: html_document
---

knitr::opts_chunk$set(echo = TRUE)
library(e1071)
library(dplyr)
library(purrr)
library(tidyr)
library(ggplot2)
library(corrplot)
library(FactoMineR)
library(VIF)
library(knitr)
library(kableExtra)
library(Hmisc)
library(pROC)
library(binr)
library(MASS)
library(pscl)
library(AER)

# read data
train = read.csv(file="data/wine-training-data.csv")
dim(train)


#transform data


#check data
summary(train) %>% kable() %>% kable_styling()

str(train)

sapply(train, function(x) sum(is.na(x))) %>% kable() %>% kable_styling()

library(UpSetR)

library(naniar)

gg_miss_fct(x = train, fct = TARGET)

train %>%
  as_shadow_upset() %>%
  upset(nsets = 24)



ntrain<-select_if(train, is.numeric)
ntrain %>%
  keep(is.numeric) %>%                     # Keep only numeric columns
  gather() %>%                             # Convert to key-value pairs
  ggplot(aes(value)) +                     # Plot the values
    facet_wrap(~ key, scales = "free") +   # In separate panels
    geom_density()

summary_metrics <- function(df){
  ###Creates summary metrics table
  metrics_only <- df[, sapply(df, is.numeric)]

  df_metrics <- psych::describe(metrics_only, quant = c(.25,.75))
  df_metrics$unique_values = rapply(metrics_only, function(x) length(unique(x)))
  df_metrics <-
    dplyr::select(df_metrics, n, unique_values, min, Q.1st = Q0.25, median, mean, Q.3rd = Q0.75,
    max, range, sd, skew, kurtosis
  )
  return(df_metrics)
}


metrics_df <- summary_metrics(train)

boxplot_data <-
  train %>%
  dplyr::select(rownames(metrics_df)[metrics_df$unique_values < 15]) %>%
  reshape2::melt(id.vars = "TARGET")

ggplot(data = boxplot_data, aes(x = factor(value), y = TARGET)) +
  geom_boxplot() +
  facet_wrap( ~ variable, scales = "free") +
  coord_flip() +
  ggthemes::theme_fivethirtyeight()
```

~~~~
trainc <- train[complete.cases(train), ]
trainc <- trainc[, !(colnames(trainc) %in% c("INDEX"))]

rcorr(as.matrix(trainc))
corrplot(cor(trainc), method="square")

library(VIM)
library(stringr)
options(scipen = 999)
missing_plot <- VIM::aggr(train,
                      numbers = T,
                      sortVars = T,
                      col = c("lightgreen", "darkred", "orange"),
                      labels=names(train),
                      ylab=c("Missing Value Counts"
                             , "Pattern"))

summary(missing_plot)

missing_plot$missings %>%
  mutate(
    pct_missing = Count / nrow(train)
    ) %>%
  arrange(-pct_missing) %>%
  filter(pct_missing > 0) %>%
  kable(digits = 3, row.names = T, caption = "Variables Missing Values")

```

## Data Preparation

```{r datapreparation}
#negative values
vars_neg_values <-
  dplyr::select(train,
            intersect(rownames(metrics_df)[metrics_df$unique_values > 15],
            rownames(metrics_df)[metrics_df$min < 0])
            )

neg_proportions <- t(apply(vars_neg_values, 2, function(x) prop.table(table(x < 0))))

data.frame(
  Var = rownames(neg_proportions),
  is_negative = neg_proportions[, 2]
) %>% arrange(-is_negative) %>%
  kable(digits = 2)

#new variables
train$BoundSulfurDioxide <- train$TotalSulfurDioxide - train$FreeSulfurDioxide

# impute data for missing values
# use column mean for calculation

train$STARS[is.na(train$STARS)] <- mean(train$STARS, na.rm=TRUE)
train$Alcohol[is.na(train$Alcohol)] <- mean(train$Alcohol, na.rm=TRUE)
train$Sulphates[is.na(train$Sulphates)] <- mean(train$Sulphates, na.rm=TRUE)
train$pH[is.na(train$pH)] <- mean(train$pH, na.rm=TRUE)
train$TotalSulfurDioxide[is.na(train$TotalSulfurDioxide)] <- mean(train$TotalSulfurDioxide, na.rm=TRUE)
train$FreeSulfurDioxide[is.na(train$FreeSulfurDioxide)] <- mean(train$FreeSulfurDioxide, na.rm=TRUE)
train$BoundSulfurDioxide[is.na(train$BoundSulfurDioxide)] <- mean(train$BoundSulfurDioxide, na.rm=TRUE)
train$Chlorides[is.na(train$Chlorides)] <- mean(train$Chlorides, na.rm=TRUE)
train$ResidualSugar[is.na(train$ResidualSugar)] <- mean(train$ResidualSugar, na.rm=TRUE)

#convert to abs for negative values
#converted to positive based upon literature

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

#new variables after abs to avoid nan and inf
train$PerVol <- train$VolatileAcidity/(train$FixedAcidity+train$VolatileAcidity)

#shift categorigal labelappeal
train$LabelAppeal <- train$LabelAppeal+2


train2<-train
train2$STARS <- as.factor(train2$STARS)



train <- train[, !(colnames(train) %in% c("INDEX"))]


#
~~~~

~~~
# #create variable
# train$new <- train$tax / (train$medv*10)
#
trainnum <- dplyr::select_if(train, is.numeric)

rcorr(as.matrix(trainnum))
corrplot(cor(trainnum), method="square")

```




## Build Models Poisson 2
```{r buildmodelspoisson}

#MODEL 1
model1 <- glm(TARGET~
FixedAcidity+VolatileAcidity+CitricAcid+ResidualSugar+Chlorides+FreeSulfurDioxide+TotalSulfurDioxide+BoundSulfurDioxide+Density+pH+Sulphates+Alcohol+as.factor(LabelAppeal)+a
s.factor(AcidIndex) + as.factor(STARS)+PerVol,data=train, family=poisson())

summary(model1)
predmodel1 <- predict(model1, type="response")
train2$pred1 <- predict(model1, type="response")

table(true = train$TARGET, pred = floor(fitted(model1))) %>% kable() %>% kable_styling()


par(mfrow=c(1,2))
hist(train2$TARGET)
hist(train2$pred1)

#plots for Model 1
par(mfrow=c(2,2))
plot(model1)

dispersiontest(model1)

#MODEL 2

model2 <- glm(TARGET~ VolatileAcidity+TotalSulfurDioxide+Alcohol+as.factor(LabelAppeal)+as.factor(AcidIndex) + as.factor(STARS)+PerVol,data=train, family=poisson())

summary(model2)
predmodel2 <- predict(model2, type="response")
train2$pred2 <- predict(model2, type="response")

table(true = train$TARGET, pred = floor(fitted(model2))) %>% kable() %>% kable_styling()

par(mfrow=c(1,2))
hist(train2$TARGET)
hist(train2$pred2)

#plots for Model 1
par(mfrow=c(2,2))
plot(model2)

dispersiontest(model2)
```



## Build Models Neg Bin Reg 2
```{r buildmodelneg}
library(MASS)
#MODEL 1
model3 <- glm.nb(TARGET~
FixedAcidity+VolatileAcidity+CitricAcid+ResidualSugar+Chlorides+FreeSulfurDioxide+TotalSulfurDioxide+BoundSulfurDioxide+Density+pH+Sulphates+Alcohol+as.factor(LabelAppeal)+a
s.factor(AcidIndex) + as.factor(STARS)+PerVol,data=train)

summary(model3)
predmodel3 <- predict(model3, type="response")
train2$pred3 <- predict(model3, type="response")

table(true = train$TARGET, pred = floor(fitted(model3))) %>% kable() %>% kable_styling()


par(mfrow=c(1,2))
hist(train2$TARGET)
hist(train2$pred3)

#plots for Model 1
par(mfrow=c(2,2))
plot(model3)

#MODEL 2

model4 <- glm.nb(TARGET~ VolatileAcidity+TotalSulfurDioxide+Alcohol+as.factor(LabelAppeal)+as.factor(AcidIndex) + as.factor(STARS)+PerVol,data=train)

summary(model4)
predmodel4 <- predict(model2, type="response")
train2$pred4 <- predict(model2, type="response")

table(true = train$TARGET, pred = floor(fitted(model4))) %>% kable() %>% kable_styling()
~~~

~~~
par(mfrow=c(1,2))
hist(train2$TARGET)
hist(train2$pred4)

#plots for Model 1
par(mfrow=c(2,2))
plot(model4)
```


## Build Models Linear 2
```{r buildmodelslinear, include=TRUE}

#MODEL 1
model5 <- lm(TARGET ~ ., data=train)
summary(model5)

par(mfrow=c(1,2))
plot(model5$residuals ~ model5$fitted.values)
plot(model5$fitted.values,train$TARGET)

par(mfrow=c(2,2))
plot(model5)

#extract variables that are significant and rerun model
sigvars <- data.frame(summary(model5)$coef[summary(model5)$coef[,4] <= .05, 4])
sigvars <- add_rownames(sigvars, "vars")
colist<-dplyr::pull(sigvars, vars)
colist <- colist[c(2:14)]

idx <- match(colist, names(train))
trainmod2 <- cbind(train[,idx], train['TARGET'])

#MODEL 2
model6<-lm(TARGET ~ ., data=trainmod2)

summary(model6)


par(mfrow=c(2,2))
plot(model6$residuals ~ model6$fitted.values)
plot(model6$fitted.values,train$TARGET)


par(mfrow=c(2,2))
plot(model6)

par(mfrow=c(1,2))
plot(model6$residuals ~ model6$fitted.values, main="New Reduced Var Model")
abline(h = 0)
plot(model5$residuals ~ model5$fitted.values, main="Orignal Model All Vars")
abline(h = 0)



```



## Select Models
```{r selectmodels}


test = read.csv(file="data/wine-evaluation-data.csv")
test2<- test
dim(test)


#new variables
test$BoundSulfurDioxide <- test$TotalSulfurDioxide - test$FreeSulfurDioxide

# impute data for missing values
# use column mean for calculation

test$STARS[is.na(test$STARS)] <- mean(test$STARS, na.rm=TRUE)
test$Alcohol[is.na(test$Alcohol)] <- mean(test$Alcohol, na.rm=TRUE)
test$Sulphates[is.na(test$Sulphates)] <- mean(test$Sulphates, na.rm=TRUE)
test$pH[is.na(test$pH)] <- mean(test$pH, na.rm=TRUE)
test$TotalSulfurDioxide[is.na(test$TotalSulfurDioxide)] <- mean(test$TotalSulfurDioxide, na.rm=TRUE)
test$FreeSulfurDioxide[is.na(test$FreeSulfurDioxide)] <- mean(test$FreeSulfurDioxide, na.rm=TRUE)
test$BoundSulfurDioxide[is.na(test$BoundSulfurDioxide)] <- mean(test$BoundSulfurDioxide, na.rm=TRUE)
test$Chlorides[is.na(test$Chlorides)] <- mean(test$Chlorides, na.rm=TRUE)
test$ResidualSugar[is.na(test$ResidualSugar)] <- mean(test$ResidualSugar, na.rm=TRUE)

#convert to abs for negative values
#converted to positive based upon literature

test$FixedAcidity <- abs(test$FixedAcidity)
test$VolatileAcidity <- abs(test$VolatileAcidity)
test$CitricAcid <- abs(test$CitricAcid)
test$ResidualSugar <- abs(test$ResidualSugar)
test$Chlorides <- abs(test$Chlorides)
~~~

```
test$FreeSulfurDioxide <- abs(test$FreeSulfurDioxide)
test$TotalSulfurDioxide <- abs(test$TotalSulfurDioxide)
test$BoundSulfurDioxide <- abs(test$BoundSulfurDioxide)
test$Sulphates <- abs(test$Sulphates)
test$Alcohol <- abs(test$Alcohol)

#new variables after abs to avoid nan and inf
test$PerVol <- test$VolatileAcidity/(test$FixedAcidity+test$VolatileAcidity)

#shift categorigal labelappeal
test$LabelAppeal <- test$LabelAppeal+2


test2<-test
test2$STARS <- as.factor(test2$STARS)



test <- test[, !(colnames(test) %in% c("INDEX"))]
test <- test[, !(colnames(test) %in% c("IN"))]

test$TARGET <- 0
test$STARS[test$STARS>2 & test$STARS <3] <- 2.04175498092412


test$TARGET <- predict(model2, newdata = test, type="response")

y_pred_num <- floor(test$TARGET)
y_pred <- factor(y_pred_num, levels=c(0, 1,2,3,4,5,6,7,8))
summary(y_pred)

rbind(round(summary(predlogit),4), round(summary(TARGET_FLAG),4)) %>% kable()

par(mfrow=c(2,2))
hist(test$TARGET)
hist(train$TARGET)
```

Appendix B: CORRELATION MATRIX

	TARGET	FixedAcidity	VolatileAcidity	CitricAcid	ResidualSugar	Chlorides	FreeSulfurDioxide	TotalSulfurDioxide	Density	pH	Sulphates	Alcohol	LabelAppeal	AcidIndex	STARS
TARGET	1	-0.01	-0.08	0	0	-0.03	0.02	0.02	-0.05	0	-0.02	0.07	0.5	-0.17	0.55
FixedAcidity	-0.01	1	0.02	0.01	-0.02	-0.01	0.02	-0.02	0.01	0	0.04	-0.01	0.01	0.15	0
VolatileAcidity	-0.08	0.02	1	-0.02	0	0.01	-0.01	0	0.01	0.01	0	0	-0.02	0.03	-0.04
CitricAcid	0	0.01	-0.02	1	-0.01	-0.03	0.01	-0.01	-0.02	0	-0.01	0.02	0.02	0.05	0.01
ResidualSugar	0	-0.02	0	-0.01	1	0	0.02	0.02	-0.01	0.02	0	-0.02	0	-0.02	0.02
Chlorides	-0.03	-0.01	0.01	-0.03	0	1	-0.02	0	0.02	-0.02	0	-0.02	-0.01	0	-0.01
FreeSulfurDioxide	0.02	0.02	-0.01	0.01	0.02	-0.02	1	0.01	-0.01	0	0.03	-0.02	0.01	-0.01	-0.02
TotalSulfurDioxide	0.02	-0.02	0	-0.01	0.02	0	0.01	1	0.02	0	0	-0.02	0	-0.02	0.02
Density	-0.05	0.01	0.01	-0.02	-0.01	0.02	-0.01	0.02	1	0	-0.01	-0.01	-0.02	0.05	-0.03
pH	0	0	0.01	0	0.02	-0.02	0	0	0	1	0.01	-0.01	0	-0.05	0
Sulphates	-0.02	0.04	0	-0.01	0	0	0.03	0	-0.01	0.01	1	0.01	0	0.03	-0.02
Alcohol	0.07	-0.01	0	0.02	-0.02	-0.02	-0.02	-0.02	-0.01	-0.01	0.01	1	0	-0.06	0.06
LabelAppeal	0.5	0.01	-0.02	0.02	0	-0.01	0.01	0	-0.02	0	0	0	1	0.01	0.32
AcidIndex	-0.17	0.15	0.03	0.05	-0.02	0	-0.01	-0.02	0.05	-0.05	0.03	-0.06	0.01	1	-0.1
STARS	0.55	0	-0.04	0.01	0.02	-0.01	-0.02	0.02	-0.03	0	-0.02	0.06	0.32	-0.1	1