Oscar Hernandez
MSDS 411
Fall 2018
Unit 3 – Wine Sales

Contents

Introduction.....3
Section 1: Initial Exploratory Data Analysis.....3-6
Section 2: Data Preparation.....6-7
Section 3: Build Models 7-12
Section 4: Select Models 12-13
Conclusion 13-14
Code 15-23

Bingo Bonus

Creative (10 Points): I decided to use random forest to impute the missing STARS values instead of creating a flag variable to display if a wine didn't have a STAR rating and created three new variables based on acidity, alcohol content and saltiness.

Introduction

The purpose of this report is to cover all the analysis that was completed for the Wine Sales data set. The Wine Sales data set contains information on over 12,000 commercially available wines. It was generated from sample cases of wine purchased by distribution companies after sampling a wine. Specifically, the data set includes 14 variables such as the alcohol content, density of a wine and rating information. The data set contains only numeric variables. Furthermore, the data set contains one response variable – TARGET which represents the amount of sample wine cases that were purchased. It is made up of discrete (count) values.
This report will cover the Exploratory Data Analysis that was conducted to better understand the data, how the data was prepared, the creation of five different models (including a Multiple Linear Regression one) and how the "best" model was determined. The rest of the models will be a Poisson or Negative Binomial regression. This analysis was done for the benefit of a large wine manufacturer that wants to understand which wine characteristics can lead to higher case sales. By understanding which wine characteristics are preferred by distributors, the manufacturer can update their offerings. Therefore, the end goal of this analysis is to develop a model that can accurately predict the number of cases ordered based upon wine characteristics.

Section 1 – Exploratory Data Analysis

Prior to building any model, it is prudent to better understand the data that will be used to train the model. Exploring data is important because we are looking to see the relationship amongst the variables included in our data, check to see if we have erroneous values and understand why we have missing values. As mentioned, our primary goal is to build a model that can predict the number of cases ordered based upon wine characteristics. Therefore, most of this report will be focused on what went into the model being developed.

As mentioned, this data set contains only numeric variables. Figure 1 is a table that displays all 14 predictor variables that are included in our data set.

Variable	Type	Variable	Type
AcidIndex	Discrete	LabelAppeal	Discrete
Alcohol	Continuous	ResidualSugar	Continuous
Chlorides	Continuous	STARS	Discrete
CitricAcid	Continuous	Sulphates	Continuous
Density	Continuous	TotalSulfurDioxide	Continuous
FixedAcidity	Continuous	VolatileAcidity	Continuous
FreeSulfurDioxide	Continuous	pH	Continuous

Figure 1: Table of Wine Sales Data

We can see from the table that 11 of the variables are continuous and 3 are discrete. However, this does not mean that the variables will stay as such to train the model. It may be prudent to transform some of these variables depending on their distribution and if the transformation provides better insight to the problem at hand. For example, the STARS variable contains discrete values, but it might make more sense to transform it to a categorical variable which could lead to a better model. Another item that we might consider is making a new Alcohol categorical variable which will have binary values based if a wine has greater than or equal some chosen alcohol content.

The next part of our analysis will include a review of all the continuous variables that are in our data set. Specifically, we want to check the distribution of these variables to see if there are some clearly erroneous values and see how much of this data subset has missing values. Figure 2 gives a robust overview of all the continuous variables.
The table suggests that we will have to come up with a way to impute for missing values since 7 of the continuous variables have missing values. The percentage of the overall values that are missing within each variable could potentially harm our model. Therefore, choosing the appropriate imputation strategy is key. Another interesting item is 9 of these variables contain negative values which is not possible since these are physical characteristics and cannot take on such states (e.g. -3.24 Citric Acid amount). Those will also have to be dealt with during the Data Preparation step. Overall, Figure 2 provides us an idea of which variables have missing values that need to be imputed along with which values are clearly wrong.

Variable	Missing	Min	Q1	Median	Mean	Q3	Max
FixedAcidity	0	-18.1	5.2	6.9	7.076	9.5	34.4
VolatilityAcidity	0	-2.79	0.13	0.28	0.3241	0.64	3.68
CitricAcid	0	-3.24	0.03	0.31	0.3084	0.58	3.86
ResidualSugar	616	-127.8	-2	3.9	5.419	15.9	141.15
Chlorides	638	-1.171	-0.031	0.046	0.0548	0.153	1.351
FreeSulfurDioxide	647	-555	0	30	30.85	70	623
TotalSulfurDioxide	682	-823	27	123	120.7	208	1057
Density	0	0.8881	0.9877	0.9945	0.9942	1.0005	1.0992
pH	395	0.48	2.96	3.2	3.208	3.470	6.130
Sulphates	1210	-3.13	0.28	0.5	0.5271	0.86	4.24
Alcohol	653	-4.7	9	10.4	10.49	12.4	26.5

Figure 2: Descriptive Statistics for Continuous Variables

Furthermore, it's important that we also review the variables that have discrete values because we may want to change the variable to a categorical one. A review of the discrete values resulted in the following takeaways:

- AcidIndex did not have any negative values or missing values.
- Approximately 44% of LabelAppeal's values were 0 but did not have missing values
- STARS variable does not contain any 0 value responses
- STARS variable had 3359 missing values which makes it the variable with the highest % missing
- Approximately 6.5% of all the wines in the data set received a 4, which is the highest STARS rating

As such, we can now conclude that *perhaps* we need to impute eight of the variables in our data set to account for missing values. Another option is to create a missing value category for any categorical variables that are developed or create binary variables where the 0 value is the non-existence of the characteristic. Additionally, from a numerical standpoint (although not displayed here), we learned that there isn't a single variable except LabelAppeal that has more than 1% of their total values as 0.

Moreover, a visual inspection of our variables can give us further insight into how they are distributed. Figure 3 shows three histograms of the variables Alcohol, AcidIndex and TARGET, respectively.

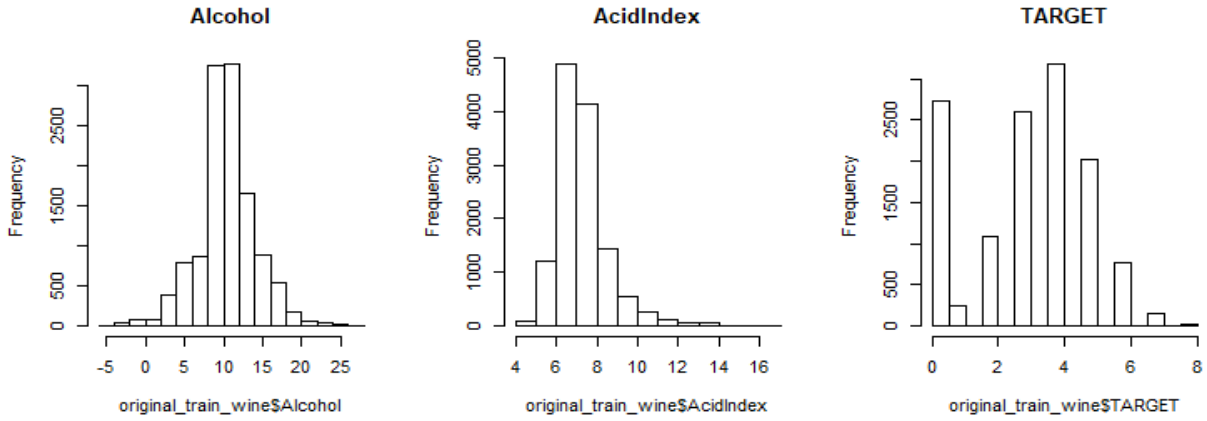

Figure 3: Histogram of Alcohol, AcidIndex and TARGET

Figure 3 shows us the distribution of a continuous predictor variable, discrete predictor variable and the discrete dependent variable. It seems the Alcohol variable follows a normal distribution with a large proportion of the values lying in the center. AcidIndex shows some signs that it has outliers given its positive skew. Lastly, the TARGET variable appears to follow a zero-inflated distribution. The values display a normal distribution aside from the large spike of 0 zero values. This is an interesting discovery because it can help us choose which type of model might be best in predicting the number of sample cases ordered.

Another quick item that we want to look at is the mean and variance of the TARGET variable. A quick computation shows that the mean value is 3.03 and the variance is 3.71. One of the assumptions of a Poisson model is that the mean and variance are equal. In this case, the values are relatively close which is important to know since it also provides us more context in deciding which type of model to utilize. If overdispersion were evident, perhaps it would be best to use a negative binomial model.

The last part of our EDA is to review the correlation between our predictor variables and the dependent variable. Figure 4 displays these correlations which shows us that LabelAppeal and STARS have the strongest correlation with TARGET. AcidIndex has the third largest correlation albeit a negative one. Interestingly enough, the other variables show weak correlation with TARGET. This could mean they might be bad predictors of the number of cases ordered. Other possible explanations include the impact that missing values and outliers are having on TARGET and the fact that TARGET is a discrete variable.

Variable	Correlation	Variable	Correlation
AcidIndex	-0.25	LabelAppeal	0.36
Alcohol	0.06	ResidualSugar	0.02
Chlorides	-0.04	STARS	0.59
CitricAcid	0.009	Sulphates	-0.04
Density	-0.04	TotalSulfurDioxide	0.05
FixedAcidity	-0.05	VolatileAcidity	-0.09
FreeSulfurDioxide	0.04	pH	-0.009

Figure 4: Correlation Table

Lastly, a visual display of the correlation is important to look at during EDA. Figure 5 shows that there doesn't appear to be a strong correlation amongst the predictor variables. The plot confirms what we already established about LabelAppeal, AcidIndex and STARS. Overall, any correlation that's present gives us preliminary insight into which variables might be strong predictors of TARGET. However, it is too soon to tell now.

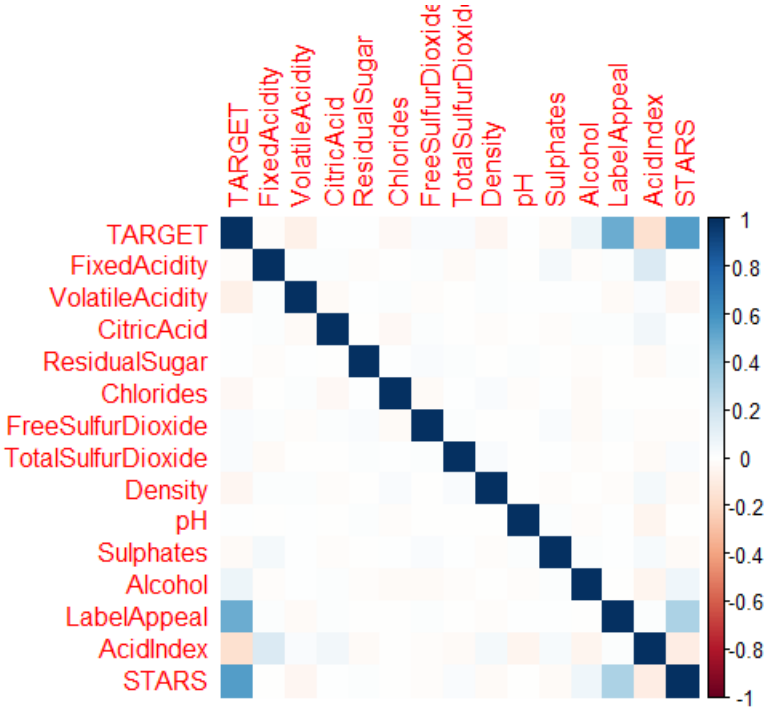

Figure 5: Correlation Plot

Model Preparation

The Model Preparation section will cover what was done on the training data set prior to fitting any models. From our EDA, we discovered that imputation of missing values is necessary. Also, the negative values need to be accounted for since wine characteristics cannot take on those states. Lastly, the decision was made to create new variables that may add to the statistical significance of our model.

The first part of the data preparation was to take all the variables that had negative values (9 to be exact) and replace the zero values with NA values. Approximately 18% of the total values from these variables were negative. The decision was made to turn them into NA values since they cannot be negative in the physical world. Therefore, they will be imputed with the rest of the missing values.

The next step in preparing the data was to impute the missing values. A popular R package called "MICE" was used to impute the missing values as opposed to simply using mean or median. The mice package uses an algorithm to replace the missing values with plausible data values. According to a popular mice tutorial, "these plausible values are drawn from a distribution specifically designed for each missing datapoint." Specifically, we used the RandomForest method within the MICE package to impute the missing values. This method was chosen because of its popularity with regards to missing values imputation.

After imputation, we determined which variables would need to stay as numeric or get converted to a categorical one. Given the distribution of LabelAppeal and STARS, the decision was made to transform them into categorical variables. This transformation was done easily since the variables contained discrete values. From the initial variables provided, there were no other categorical transformations made.

From EDA, we did notice some variables contained outliers. Furthermore, secondary research was completed on physical wine characteristics to determine what range of values seem reasonable. After getting a better understanding of wine, it was decided to winsorize values that were above the 98th percentile for all the numeric variables. Winsorizing was not done for values less than the 2nd percentile since this would remove 0 values which doesn't make sense.

After winsorizing certain variables, the next step was to create new variables. Based on correlation, we decided that AcidIndex may be a good predictor of TARGET. AcidIndex tests total acidity of a particular wine and the belief is that a higher AcidIndex would negatively impact sample cases ordered. The decision was made to create a AcidWine_FLAG variable that has binary values – AcidWine or NonAcidWine. A similar logic was applied to the creation of two other variables, SaltyWine_FLAG and StrongWine_FLAG, which simply show if a wine has a high level of chlorides or alcohol content, respectively.

The last data preparation activity includes variable transformation. Based on our EDA, the decision was made to do a square root transformation on the following variables: VolatileAcidity, ResidualSugar, CitricAcid, Chlorides, FreeSulfurDioxide and TotalSulfurDioxide. This transformation was done to change the distribution of these variables.

Build Model

This section of the report will cover the five regression models that were built and a discussion of some of their key aspects. Figure 6, 7, 8, 9 and 10 show the model summary data for the five models that were built – "Stepwise MLR Model", "Poisson Model", "Negative Binomial Model", "Zero-Inflated Poisson Model" and "Zero-Inflated Negative Binomial Model."

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	7.75	0.52	14.88	< 2e-16
Density	-0.89	0.50	-1.78	0.07
STARSOne_Star	-2.62	0.06	-41.15	< 2e-16
STARSThree_Stars	-0.67	0.07	-10.21	< 2e-16
STARSTwo_Stars	-1.38	0.06	-21.85	< 2e-16
LabelAppealNeg_Two	-0.47	0.07	-6.54	0.00
LabelAppealPos_One	1.06	0.04	26.81	< 2e-16
LabelAppealPos_Two	1.63	0.07	21.85	< 2e-16
LabelAppealz_Indifferent	0.56	0.03	16.42	< 2e-16
AcidIndex	-0.37	0.02	-22.65	< 2e-16
VolatileAcidity	-0.20	0.03	-7.73	0.00

AcidicWine_Flagz_AcidicWine	0.26	0.04	6.45	0.00
TotalSulfurDioxide	0.00	0.00	5.21	0.00
Alcohol	0.02	0.01	4.74	0.00
Chlorides	-0.74	0.15	-5.10	0.00
SaltyWine_Flagz_SaltyWine	0.24	0.07	3.60	0.00
CitricAcid	0.08	0.02	3.36	0.00
Sulphates	-0.07	0.02	-3.22	0.00
FreeSulfurDioxide	0.00	0.00	3.09	0.00
pH	-0.05	0.02	-2.22	0.03
StrongWine_Flagz_StrongWine	-0.09	0.06	-1.55	0.12

Figure 6: Stepwise Model

The Stepwise Model was calculated using the AIC statistic in Stepwise fashion. Figure 6 provides us a view of all the variables, their respective coefficients and statistical significance. We can see that the majority of the variables are statistically significant based on an alpha of 0.05. Notably, we can see that STARS variable is statistically significant. This makes sense given their high correlation with TARGET and with the belief that the ratings a wine received is probably a good indicator of ordering or not ordering a sample case. Furthermore, the LabelAppeal variable show statistical significance which follows similar logic to the STARS variable. It's important to highlight that LabelAppealPos_Two has the largest positive coefficient and STARSOne_Star has the largest negative coefficient. Both of these results make sense. Lastly, it's interesting to see that one of our newly created variables StrongWine_FLAG is not statistically significant which means that a high level of alcohol content in a wine doesn't have a material impact on the amount of cases ordered.

	Estimate	**Std. Error**	**z value**	**Pr(>|z|)**
(Intercept)	2.67	0.20	13.26	< 2e-16
STARSOne_Star	-0.79	0.02	-38.21	< 2e-16
STARSThree_Stars	-0.12	0.02	-6.32	0.00
STARSTwo_Stars	-0.30	0.02	-15.43	< 2e-16
LabelAppealNeg_Two	-0.28	0.04	-7.50	0.00
LabelAppealPos_One	0.36	0.02	23.00	< 2e-16
LabelAppealPos_Two	0.49	0.02	19.67	< 2e-16
LabelAppealz_Indifferent	0.22	0.01	15.38	< 2e-16
Density	-0.30	0.19	-1.55	0.12
Alcohol	0.01	0.00	4.03	0.00
AcidIndex	-0.15	0.01	-20.31	< 2e-16
pH	-0.02	0.01	-2.07	0.04
TotalSulfurDioxide	0.01	0.00	6.33	0.00
Chlorides	-0.46	0.06	-7.44	0.00
VolatileAcidity	-0.13	0.02	-7.23	0.00
CitricAcid	0.08	0.02	4.95	0.00
Sulphates	-0.02	0.01	-2.86	0.00

AcidicWine_Flagz_AcidicWine	0.13	0.02	8.00	0.00
SaltyWine_Flagz_SaltyWine	0.18	0.03	6.24	0.00
StrongWine_Flagz_StrongWine	-0.04	0.02	-1.72	0.08
ResidualSugar	0.00	0.00	1.86	0.06
FreeSulfurDioxide	0.00	0.00	4.25	0.00

Figure 7: Poisson Model

The Poisson Model was determined manually as opposed to using a systematic fashion such as Stepwise. There was no specific rule used to manually develop this model. This model was created purely on secondary research that was completed on popular wine characteristics and historical wine sales that we believe would be good predictors of sample wine cases ordered.

Figure 7 provides us a view of all the variables, their respective coefficients and statistical significance. We can see that most of the variables have a statistically significant coefficient. Reviewing the coefficients, we see STARS and LabelAppeal variables are the largest just like the Stepwise model. Once again this makes sense for the same reason already described. However, an interesting development is how the Chlorides coefficient is just as large as some of the STARS or LabelAppeal. This wasn't the case in the Stepwise Model. Furthermore, we see once again that the StrongWine_FLAG and Density variables are not statistically significant. Overall, the LabelAppealPos_Two coefficient has the largest, positive impact in the Poisson Model just as the Stepwise Model.

	Estimate	**Std. Error**	**z value P**	**r(>|z|)**
(Intercept)	2.60	0.06	44.09	< 2e-16
STARSOne_Star	-0.77	0.01	-125.75	< 2e-16
STARSThree_Stars	-0.13	0.01	-21.74	< 2e-16
STARSTwo_Stars	-0.30	0.01	-52.57	< 2e-16
LabelAppealNeg_Two	-0.29	0.01	-25.57	< 2e-16
LabelAppealPos_One	0.36	0.00	79.65	< 2e-16
LabelAppealPos_Two	0.47	0.01	64.16	< 2e-16
LabelAppealz_Indifferent	0.22	0.00	52.17	< 2e-16
Density	-0.18	0.06	-3.19	0.00
Alcohol	0.01	0.00	14.35	< 2e-16
AcidIndex	-0.15	0.00	-66.80	< 2e-16
pH	-0.01	0.00	-6.20	0.00
TotalSulfurDioxide	0.01	0.00	19.14	< 2e-16
Chlorides	-0.47	0.02	-25.47	< 2e-16
VolatileAcidity	-0.13	0.01	-25.20	< 2e-16
CitricAcid	0.08	0.00	16.61	< 2e-16
Sulphates	-0.02	0.00	-6.66	0.00
AcidicWine_Flagz_AcidicWine	0.14	0.00	29.55	< 2e-16
SaltyWine_Flagz_SaltyWine	0.18	0.01	21.19	< 2e-16
StrongWine_Flagz_StrongWine	-0.04	0.01	-5.73	0.00

Figure 8: Negative Binomial Model

The Negative Binomial Model was determined in the same fashion as the Poisson Model. However, it doesn't include two variables – ResidualSugar and FreeSulfurDioxide. Nevertheless, the coefficients are nearly identical compared to the Poisson Model. Therefore, a review of this model won't reveal more insights than already gleaned from the Poisson Model. The only interesting item is that all the variables in the Negative Binomial Model were statistically significant which wasn't the case in the Poisson Model.

Count model coefficients (poisson with log link)

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.51	0.21	7.14	0.00
STARSOne_Star	-0.30	0.02	-13.90	< 2e-16
STARSThree_Stars	-0.09	0.02	-4.65	0.00
STARSTwo_Stars	-0.19	0.02	-9.46	< 2e-16
LabelAppealNeg_Two	-0.45	0.04	-10.87	< 2e-16
LabelAppealPos_One	0.52	0.02	31.20	< 2e-16
LabelAppealPos_Two	0.69	0.03	26.45	< 2e-16
LabelAppealz_Indifferent	0.32	0.02	20.67	< 2e-16
Density	-0.23	0.20	-1.17	0.24
Alcohol	0.01	0.00	5.46	0.00
AcidIndex	-0.02	0.01	-2.64	0.01
pH	0.01	0.01	0.62	0.54
TotalSulfurDioxide	0.00	0.00	-0.64	0.52
Chlorides	-0.19	0.06	-2.90	0.00
VolatileAcidity	-0.04	0.02	-2.14	0.03
CitricAcid	0.02	0.02	0.92	0.36
Sulphates	0.00	0.01	0.30	0.77
AcidicWine_Flagz_AcidicWine	0.00	0.02	0.21	0.83
SaltyWine_Flagz_SaltyWine	0.08	0.03	2.55	0.01
StrongWine_Flagz_StrongWine	-0.05	0.02	-2.21	0.03
ResidualSugar	0.00	0.00	0.07	0.94
FreeSulfurDioxide	0.00	0.00	0.92	0.36
Zero-inflation model coefficients (binomial with logit link)				
	Estimate	**Std. Error**	**z value**	**Pr(>|z|)**
(Intercept)	-10.57	1.25	-8.43	< 2e-16
STARSOne_Star	3.78	0.34	11.03	< 2e-16
STARSThree_Stars	0.55	0.36	1.52	0.13
STARSTwo_Stars	1.90	0.34	5.52	0.00
LabelAppealNeg_Two	-1.41	0.38	-3.75	0.00
LabelAppealPos_One	1.20	0.10	12.10	< 2e-16
LabelAppealPos_Two	1.56	0.17	9.29	< 2e-16
LabelAppealz_Indifferent	0.70	0.09	7.95	0.00
Density	0.69	1.16	0.60	0.55

Alcohol	0.02	0.01	1.95	0.05
AcidIndex	0.59	0.03	17.51	< 2e-16
pH	0.17	0.05	3.64	0.00
TotalSulfurDioxide	-0.05	0.01	-8.10	0.00
Chlorides	1.74	0.35	4.96	0.00
VolatileAcidity	0.58	0.10	5.97	0.00
CitricAcid	-0.51	0.10	-5.01	0.00
Sulphates	0.19	0.05	3.71	0.00
AcidicWine_Flagz_AcidicWine	-0.40	0.09	-4.41	0.00
SaltyWine_Flagz_SaltyWine	-0.65	0.17	-3.88	0.00
StrongWine_Flagz_StrongWine	-0.14	0.13	-1.06	0.29
ResidualSugar	-0.02	0.01	-1.82	0.07
FreeSulfurDioxide	-0.03	0.01	-3.70	0.00

Figure 9: ZIP Model

The ZIP Model was determined in the same fashion as the Poisson Model. This model was created after reviewing TARGET variable and noticing the large amount of zero values in the distribution. We noticed that for the count part of the model, that the majority of the variables are not statistically significant. This is a meaningful departure from the other models that have been presented thus far and it's something that we will keep in mind when choosing the "best" model. The STARS and LabelAppeal variables maintain their statistical significance as well. An interesting development is the magnitude in which the LabelAppealPos_Two coefficient changes in this model. Its impact is much larger than in the Poisson Model which is noteworthy. The STARSOne_Star coefficient also has a larger impact compared to the Poisson Model.

Count model coefficients (negbin with log link)

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.53	0.21	7.22	0.00
STARSOne_Star	-0.30	0.02	-13.87	< 2e-16
STARSThree_Stars	-0.09	0.02	-4.62	0.00
STARSTwo_Stars	-0.19	0.02	-9.42	< 2e-16
LabelAppealNeg_Two	-0.45	0.04	-10.87	< 2e-16
LabelAppealPos_One	0.53	0.02	31.23	< 2e-16
LabelAppealPos_Two	0.69	0.03	26.48	< 2e-16
LabelAppealz_Indifferent	0.32	0.02	20.69	< 2e-16
Density	-0.24	0.20	-1.18	0.24
Alcohol	0.01	0.00	5.44	0.00
AcidIndex	-0.02	0.01	-2.70	0.01
pH	0.01	0.01	0.61	0.54
TotalSulfurDioxide	0.00	0.00	-0.60	0.55
Chlorides	-0.19	0.06	-2.87	0.00
VolatileAcidity	-0.04	0.02	-2.15	0.03
CitricAcid	0.02	0.02	0.92	0.36

Sulphates	0.00	0.01	0.28	0.78
AcidicWine_Flagz_AcidicWine	0.00	0.02	0.24	0.81
SaltyWine_Flagz_SaltyWine	0.08	0.03	2.53	0.01
StrongWine_Flagz_StrongWine	-0.05	0.02	-2.20	0.03
Log(theta)	12.14	3.64	3.34	0.00
Zero-inflation model coefficients (binomial with logit link)				
	Estimate	**Std. Error**	**z value**	**Pr(>|z|)**
(Intercept)	-10.92	1.25	-8.72	< 2e-16
STARSOne_Star	3.80	0.35	10.95	< 2e-16
STARSThree_Stars	0.57	0.37	1.55	0.12
STARSTwo_Stars	1.91	0.35	5.50	0.00
LabelAppealNeg_Two	-1.44	0.38	-3.77	0.00
LabelAppealPos_One	1.20	0.10	12.08	< 2e-16
LabelAppealPos_Two	1.55	0.17	9.26	< 2e-16
LabelAppealz_Indifferent	0.69	0.09	7.92	0.00
Density	0.68	1.15	0.59	0.56
Alcohol	0.03	0.01	2.12	0.03
AcidIndex	0.60	0.03	17.79	< 2e-16
pH	0.17	0.05	3.60	0.00
TotalSulfurDioxide	-0.05	0.01	-8.47	< 2e-16
Chlorides	1.77	0.35	5.05	0.00
VolatileAcidity	0.59	0.10	6.08	0.00
CitricAcid	-0.52	0.10	-5.18	0.00
Sulphates	0.19	0.05	3.76	0.00
AcidicWine_Flagz_AcidicWine	-0.40	0.09	-4.43	0.00
SaltyWine_Flagz_SaltyWine	-0.67	0.17	-3.99	0.00
StrongWine_Flagz_StrongWine	-0.14	0.13	-1.07	0.28

Figure 10: ZINB Model

The last model that was developed was the ZINB Model which was created in the same fashion as the Poisson Model. The coefficients in this model are nearly identical to the ZIP Model therefore further examination of this model won't provide us any new insights. An interesting point is that most of the variables in this model are not statistically significant much like the ZIP Model. This is a departure from the Negative Binomial Model which had coefficients that were all statistically significant.

Select Models

The final section of this report will review which model was determined to be our "best" model. In order to make such a validation, we calculated multiple metrics that are commonly used to compare models using training data along with completing a Vuong Test. Figure 11 displays the metrics that were calculated with respect to each model.

	AIC	Rank	BIC	Rank	-2*Log Likelihood	Rank
Stepwise MLR Model	46793	3rd	46957	3rd	46874	3rd
Poisson Model	48448	4th	48612	4th	48404	4th
Negative Binomial Model	552716	5th	552873	5th	552674	5th
ZIP Model	43383	1st	43711	2nd	43295	1st
ZINB Model	43397	2nd	43702	1st	43315	2nd

Figure 11: Model Comparison (Train Data)

Figure 11 ranks where each model lies with regards to each metric. The metrics used are to provide a comparison of model accuracy. The data suggests that the ZIP Model is the most accurate model given that it ranked 1st in nearly every single category followed by the ZINB Model. This isn't surprising since the distribution of the TARGET variable was zero-inflated. Furthermore, we see that the Negative Binomial Model is by far the worse model given that its validation metrics weren't even competitive. Surprisingly, we see that the Stepwise MLR Model was not too far off in terms of how its validation metrics when compared to either the ZIP or ZINB Model.
Moreover, from the previous section we noticed that the ZIP model had several non-statistically significant variables which is concerning. However, when reviewing its coefficients, the ones that were statistically significant were similar when compared to other models. Given the distribution of the TARGET Variable, we will not consider the Stepwise MLR as a candidate for the "best" model. Also, given how poorly the Negative Binomial Model performed with regards to validation metrics, we will also discard this model. Therefore, to determine the "best" model we decided to conduct a Vuong Test between the Poisson and ZIP Model. The results show that the ZIP Model is better based on the test. Therefore based on the Vuong Test and other items discussed in this report, we have concluded that the ZIP model is the "best" model.

Conclusion

The purpose of this report was to explain the analysis and logic behind developing several models to predict the number of sample cases that would be ordered based on wine characteristics. The wine data gave us a ton of information about the physical characteristics of wine which admittedly didn't seem like it would be helpful given that preference for wine varies across regions. Nonetheless, the data proved useful in constructing our model.

The Exploratory Data Analysis section was the initial groundwork that allowed us to first notice some of the subsequent work that would have to be done with regards to imputing missing/negative values, getting an idea of which variables may need to get transformed/winsorized and creating new variables. The subsequent parts explained the logic in preparing the training data, how each model was built and ultimately what criteria was used to select the model that we believe will do the best job at predicting the number of sample cases ordered.

From this exercise, it appears that using a zero-inflated model is preferred when the dependent variable follows a zero-inflated distribution. Furthermore, the coefficients in our "best" model reaffirmed in what we initially believed would be good predictors of wine sales – STARS and LabelAppeal. From our analysis,

we learned that having high expert ratings and popular label designs are strong influencers on wine sales. Initially, we did believe that AcidIndex would be a good predictor variable since it had one of the highest correlations with the TARGET variable but based on its coefficient, it wasn't as impactful in terms of magnitude as the other highly correlated variables. We also learned that having high alcohol content doesn't translate to more sample cases being ordered. The StrongWine_FLAG variable provided us that insight. Overall, our "best" model seems to be valuable in predicting the number of sample wine cases ordered.

R Code

```
#####Import the necessary packages
library(mice)
library(MASS)
library(corrplot)
library(gmodels)
library(robustHD)
library(pscl)
#####

##### Set working directory and load test data
setwd("C:/Users/herna_000/Desktop")
train_wine <- read.csv("Wine_Training.csv", header=TRUE)
original_train_wine <- train_wine
#####

############ PART 1: DATA EXPLORATION
#############################################################
str(original_train_wine)
summary(original_train_wine)

hist(original_train_wine$TARGET) #zero inflated; otherwise normal
boxplot(original_train_wine$TARGET)

CrossTable(original_train_wine$LabelAppeal)
CrossTable(original_train_wine$STARS)

par(mfrow=c(1,3))
hist(original_train_wine$Alcohol, main="Alcohol")
hist(original_train_wine$AcidIndex, main="AcidIndex")
hist(original_train_wine$TARGET,main="TARGET")
par(mfrow=c(1,1))

hist(original_train_wine$FixedAcidity) #outliers; otherwise normal; lot of values between 5 - 10
boxplot(original_train_wine$FixedAcidity)

hist(original_train_wine$VolatileAcidity) #outliers; otherwise normal; lots of values between 0 - 1
boxplot(original_train_wine$VolatileAcidity)

hist(original_train_wine$CitricAcid)  #outliers; otherwise normal; lots of values around 0 - 0.5
boxplot(original_train_wine$CitricAcid)

hist(original_train_wine$ResidualSugar) #outliers; otherwise normal; lots of values around 0 - 20
boxplot(original_train_wine$ResidualSugar)

hist(original_train_wine$Chlorides) #outliers; otherwise normal; lots of values around 0 - 0.2
boxplot(original_train_wine$Chlorides)
```

```
hist(original_train_wine$FreeSulfurDioxide)  #outliers; otherwise normal; lots of values around 0 - 100
boxplot(original_train_wine$FreeSulfurDioxide)

hist(original_train_wine$TotalSulfurDioxide)  #outliers; otherwise normal; lots of values around 0 - 200
boxplot(original_train_wine$TotalSulfurDioxide)

hist(original_train_wine$Density)  #outliers; otherwise normal; lots values around 0.98 - 1
boxplot(original_train_wine$Density)

hist(original_train_wine$pH)  #outliers; otherwise normal; lots of values 3 - 3.5
boxplot(original_train_wine$pH)

hist(original_train_wine$Sulphates)  #outliers; otherwise normal; lots of values around 0 - 1
boxplot(original_train_wine$Sulphates)

hist(original_train_wine$Alcohol)  #outliers; othewise normal; lotsof values around 10
boxplot(original_train_wine$Alcohol)

hist(original_train_wine$AcidIndex)  #some outliers; not normal; positive skew
boxplot(original_train_wine$AcidIndex)

#Scatterplot
plot(original_train_wine$FixedAcidity, original_train_wine$TARGET)
plot(original_train_wine$VolatileAcidity, original_train_wine$TARGET)
plot(original_train_wine$CitricAcid, original_train_wine$TARGET)
plot(original_train_wine$ResidualSugar, original_train_wine$TARGET)


#Correlation
cor(original_train_wine$FixedAcidity, original_train_wine$TARGET)
cor(original_train_wine$VolatileAcidity, original_train_wine$TARGET)
cor(original_train_wine$CitricAcid, original_train_wine$TARGET)
cor(original_train_wine$Density, original_train_wine$TARGET)
cor(original_train_wine$AcidIndex, original_train_wine$TARGET)
cor(original_train_wine$ResidualSugar, original_train_wine$TARGET, use = "na.or.complete")
cor(original_train_wine$Chlorides, original_train_wine$TARGET, use = "na.or.complete")
cor(original_train_wine$FreeSulfurDioxide, original_train_wine$TARGET, use = "na.or.complete")
cor(original_train_wine$TotalSulfurDioxide, original_train_wine$TARGET, use = "na.or.complete")
cor(original_train_wine$pH, original_train_wine$TARGET, use = "na.or.complete")
cor(original_train_wine$Sulphates, original_train_wine$TARGET, use = "na.or.complete")
cor(original_train_wine$Alcohol, original_train_wine$TARGET, use = "na.or.complete")
cor(original_train_wine$LabelAppeal, original_train_wine$TARGET, use = "na.or.complete")
cor(original_train_wine$STARS, original_train_wine$TARGET, use = "na.or.complete")

wine_train_corr <- cor(original_train_wine[, !names(original_train_wine) %in% c("INDEX")],
use="na.or.complete")
corrplot(wine_train_corr,method="color")
```

```
mean(original_train_wine$TARGET)
var(original_train_wine$TARGET)

############ PART 2: DATA PREPARATION
############################################################
removed_columns <- data.frame(train_wine$INDEX)

#Turn negative values into missing values
train_wine$Alcohol[train_wine$Alcohol<0] <- NA
train_wine$Chlorides[train_wine$Chlorides<0] <- NA
train_wine$CitricAcid[train_wine$CitricAcid<0] <- NA
train_wine$FixedAcidity[train_wine$FixedAcidity<=0] <- NA
train_wine$FreeSulfurDioxide[train_wine$FreeSulfurDioxide<0] <- NA
train_wine$ResidualSugar[train_wine$ResidualSugar<0] <- NA
train_wine$Sulphates[train_wine$Sulphates<0] <- NA
train_wine$TotalSulfurDioxide[train_wine$TotalSulfurDioxide<0] <- NA
train_wine$VolatileAcidity[train_wine$VolatileAcidity<0] <- NA

#Impute missing values using MICE package
train__wine_temp1 <- mice(train_wine[, !names(train_wine) %in% c("INDEX")], m =5, maxit=10, meth =
"rf", seed = 500)
train_wine_imp1 <-complete(train__wine_temp1)  #restart here if needed
train_wine_imp1 <- cbind(removed_columns, train_wine_imp1)

colnames(train_wine_imp1)[colnames(train_wine_imp1)=="train_wine.INDEX"] <- "INDEX"

str(train_wine_imp1)
summary(train_wine_imp1)

#Convert certain variables into categorical variables
CrossTable(train_wine_imp1$LabelAppeal)

train_wine_imp1$LabelAppeal[train_wine_imp1$LabelAppeal==-2] <- "Neg_Two"
train_wine_imp1$LabelAppeal[train_wine_imp1$LabelAppeal==-1] <- "Neg_One"
train_wine_imp1$LabelAppeal[train_wine_imp1$LabelAppeal==0] <- "z_Indifferent"
train_wine_imp1$LabelAppeal[train_wine_imp1$LabelAppeal==1] <- "Pos_One"
train_wine_imp1$LabelAppeal[train_wine_imp1$LabelAppeal==2] <- "Pos_Two"

train_wine_imp1$LabelAppeal <- as.factor(train_wine_imp1$LabelAppeal)

#Recode STARS variable; Turn NA's into "Missing"
CrossTable(train_wine_imp1$STARS)
train_wine_imp1$STARS <- as.character(train_wine_imp1$STARS)

train_wine_imp1$STARS[train_wine_imp1$STARS==1] <- "One_Star"
train_wine_imp1$STARS[train_wine_imp1$STARS==2] <- "Two_Stars"
train_wine_imp1$STARS[train_wine_imp1$STARS==3] <- "Three_Stars"
```

```
train_wine_imp1$STARS[train_wine_imp1$STARS==4] <- "Four_Stars"

train_wine_imp1$STARS <- as.factor(train_wine_imp1$STARS)

#Check percentile values and possibly do windsorizing
fa_high <- quantile(train_wine_imp1$FixedAcidity, c(.98))
va_high <- quantile(train_wine_imp1$VolatileAcidity, c(.98))
rs_high <- quantile(train_wine_imp1$ResidualSugar, c(.98))
ca_high <- quantile(train_wine_imp1$CitricAcid, c(.98))
c_high <- quantile(train_wine_imp1$Chlorides, c(.98))
fsd_high <- quantile(train_wine_imp1$FreeSulfurDioxide, c(.98))
tsd_high <- quantile(train_wine_imp1$TotalSulfurDioxide, c(.98))
d_high <- quantile(train_wine_imp1$Density, c(.98))
ph_high <- quantile(train_wine_imp1$pH, c(.98))
s_high <- quantile(train_wine_imp1$Sulphates, c(.98))
a_high <- quantile(train_wine_imp1$Alcohol, c(.98))
ai_high <- quantile(train_wine_imp1$AcidIndex, c(.98))

train_wine_imp1$FixedAcidity[train_wine_imp1$FixedAcidity > fa_high] <- fa_high
train_wine_imp1$VolatileAcidity[train_wine_imp1$VolatileAcidity > va_high] <- va_high
train_wine_imp1$ResidualSugar[train_wine_imp1$ResidualSugar>rs_high] <- rs_high
train_wine_imp1$CitricAcid[train_wine_imp1$CitricAcid>ca_high] <- ca_high
train_wine_imp1$Chlorides[train_wine_imp1$Chlorides>c_high] <- c_high
train_wine_imp1$FreeSulfurDioxide[train_wine_imp1$FreeSulfurDioxide>fsd_high] <- fsd_high
train_wine_imp1$TotalSulfurDioxide[train_wine_imp1$TotalSulfurDioxide>tsd_high] <- tsd_high
train_wine_imp1$pH[train_wine_imp1$pH>ph_high] <- ph_high
train_wine_imp1$Sulphates[train_wine_imp1$Sulphates>s_high] <- s_high
train_wine_imp1$Alcohol[train_wine_imp1$Alcohol>a_high] <- a_high
train_wine_imp1$AcidIndex[train_wine_imp1$AcidIndex>ai_high] <- ai_high

#Create new variables
train_wine_imp1$AcidicWine_Flag <- train_wine_imp1$AcidIndex
train_wine_imp1$AcidicWine_Flag <- ifelse(train_wine_imp1$AcidicWine_Flag >7, "z_AcidicWine",
"NonAcidicWine")
train_wine_imp1$AcidicWine_Flag <- as.factor(train_wine_imp1$AcidicWine_Flag)

train_wine_imp1$SaltyWine_Flag <- train_wine_imp1$Chlorides
train_wine_imp1$SaltyWine_Flag <- ifelse(train_wine_imp1$SaltyWine_Flag > 0.2, "z_SaltyWine",
"NonSaltyWine")
train_wine_imp1$SaltyWine_Flag <- as.factor(train_wine_imp1$SaltyWine_Flag)

train_wine_imp1$StrongWine_Flag <- train_wine_imp1$Alcohol
train_wine_imp1$StrongWine_Flag <- ifelse(train_wine_imp1$StrongWine_Flag > 15, "z_StrongWine",
"WeakWine")
train_wine_imp1$StrongWine_Flag <- as.factor(train_wine_imp1$StrongWine_Flag)

#Square Root Transformation
train_wine_imp1$VolatileAcidity <- sqrt(train_wine_imp1$VolatileAcidity)
```

```
train_wine_imp1$ResidualSugar <- sqrt(train_wine_imp1$ResidualSugar)
train_wine_imp1$CitricAcid <- sqrt(train_wine_imp1$CitricAcid)
train_wine_imp1$Chlorides <- sqrt(train_wine_imp1$Chlorides)
train_wine_imp1$FreeSulfurDioxide <- sqrt(train_wine_imp1$FreeSulfurDioxide)
train_wine_imp1$TotalSulfurDioxide <- sqrt(train_wine_imp1$TotalSulfurDioxide)

#Check variable distribtuions
hist(train_wine_imp1$FixedAcidity)
hist(train_wine_imp1$VolatileAcidity) #sq
hist(train_wine_imp1$ResidualSugar) #sq
hist(train_wine_imp1$CitricAcid) #sq
hist(train_wine_imp1$Chlorides)#sq
hist(train_wine_imp1$FreeSulfurDioxide) #sq
hist(train_wine_imp1$TotalSulfurDioxide) #sq
hist(train_wine_imp1$Density)
hist(train_wine_imp1$pH)
hist(train_wine_imp1$Sulphates)
hist(train_wine_imp1$Alcohol)
hist(train_wine_imp1$AcidIndex)

str(train_wine_imp1)
summary(train_wine_imp1)

exp(-0.79)
############ PART 3: BUILD MODELS
#####################################################################

#####MLR Model - Using Stepwise Variable Selection #####
#Need to specifiy the upper model and lower models
# Define the upper model as the FULL model
upper.lm <- lm(TARGET ~ .,data=train_wine_imp1[, !names(train_wine_imp1) %in% "INDEX"])

# Define the lower model as the Intercept model
lower.lm <- lm(TARGET ~ 1,data=train_wine_imp1[, !names(train_wine_imp1) %in% "INDEX"])

# Need a simple linear regression model to initialize stepwise selection
initialize_stepwise.lm <- lm(TARGET ~ Density,data=train_wine_imp1[, !names(train_wine_imp1) %in%
"INDEX"])

#Create stepwise.lm
stepwise.lm <- stepAIC(object=initialize_stepwise.lm,scope=list(upper=formula(upper.lm),lower=~1),
direction=c('both'))
summary(stepwise.lm)

#####Poisson Model ######
poisson.glm <- glm(TARGET ~ STARS+ LabelAppeal+ Density+ Alcohol+AcidIndex+ pH+
TotalSulfurDioxide+ Chlorides+ VolatileAcidity+
```

```
              CitricAcid + Sulphates + AcidicWine_Flag + SaltyWine_Flag + StrongWine_Flag +
ResidualSugar + FreeSulfurDioxide,
            family = "poisson"(link="log"), data=train_wine_imp1)
summary(poisson.glm)

#####Negative Binomial Model #####
NBR.glm <- glm.nb(TARGET~ STARS+ LabelAppeal+ Density+ Alcohol+AcidIndex+ pH+
TotalSulfurDioxide+ Chlorides+ VolatileAcidity+
              CitricAcid + Sulphates + AcidicWine_Flag + SaltyWine_Flag + StrongWine_Flag, ResidualSugar
+ FreeSulfurDioxide,
            data=train_wine_imp1)
summary(NBR.glm)

#####ZI Poisson Model#####
zip.glm<-zeroinfl(TARGET~ STARS+ LabelAppeal+ Density+ Alcohol+AcidIndex+  pH+ TotalSulfurDioxide+
Chlorides+ VolatileAcidity+
              CitricAcid + Sulphates + AcidicWine_Flag + SaltyWine_Flag + StrongWine_Flag +ResidualSugar
+ FreeSulfurDioxide,
            data=train_wine_imp1)
summary(zip.glm)

####ZI NB Model#####
zinb.glm <- zeroinfl(TARGET~ STARS+ LabelAppeal+ Density+ Alcohol+AcidIndex+ pH+
TotalSulfurDioxide+ Chlorides+ VolatileAcidity+
                CitricAcid+ Sulphates + AcidicWine_Flag + SaltyWine_Flag + StrongWine_Flag,
              dist="negbin", EM=TRUE, data=train_wine_imp1)
summary(zinb.glm)


############ PART 4: SELECT MODELS
#####################################################################

AIC(stepwise.lm)
BIC(stepwise.lm)
print(-2*logLik(stepwise.lm, REML = TRUE))

AIC(poisson.glm)
BIC(poisson.glm)
print(-2*logLik(poisson.glm), REML=TRUE)

AIC(NBR.glm)
BIC(NBR.glm)
print(-2*logLik(NBR.glm), REML=TRUE)

AIC(zip.glm)
AIC(zip.glm, k=log(nrow(train_wine_imp1))) #BIC
print(-2*logLik(zip.glm), REML=TRUE)
```

```
AIC(zinb.glm)
AIC(zinb.glm, k=log(nrow(train_wine_imp1))) #BIC
print(-2*logLik(zinb.glm), REML=TRUE)

vuong(poisson.glm, zip.glm)
vuong(NBR.glm, zinb.glm)


##########################STAND ALONE SCORING
PROGRAM#################################
original_test_wine <- read.csv("Wine_Test.csv", header = TRUE)
test_wine <- original_test_wine

removed_columns_test <- data.frame(test_wine$INDEX)

#Turn negative values into missing values
test_wine$Alcohol[test_wine$Alcohol<0] <- NA
test_wine$Chlorides[test_wine$Chlorides<0] <- NA
test_wine$CitricAcid[test_wine$CitricAcid<0] <- NA
test_wine$FixedAcidity[test_wine$FixedAcidity<0] <- NA
test_wine$FreeSulfurDioxide[test_wine$FreeSulfurDioxide<0] <- NA
test_wine$ResidualSugar[test_wine$ResidualSugar<0] <- NA
test_wine$Sulphates[test_wine$Sulphates<0] <- NA
test_wine$TotalSulfurDioxide[test_wine$TotalSulfurDioxide<0] <- NA
test_wine$VolatileAcidity[test_wine$VolatileAcidity<0] <- NA

#Impute missing values using MICE package
test_wine_temp1 <- mice(test_wine[, !names(test_wine) %in% c("INDEX")], m =5, maxit=10, meth = "rf",
seed = 500)
test_wine_imp1 <-complete(test_wine_temp1)  #restart here if needed
test_wine_imp1 <- cbind(removed_columns_test, test_wine_imp1)

colnames(test_wine_imp1)[colnames(test_wine_imp1)=="test_wine.INDEX"] <- "INDEX"

#Convert certain variables into categorical variables

test_wine_imp1$LabelAppeal[test_wine_imp1$LabelAppeal==-2] <- "Neg_Two"
test_wine_imp1$LabelAppeal[test_wine_imp1$LabelAppeal==-1] <- "Neg_One"
test_wine_imp1$LabelAppeal[test_wine_imp1$LabelAppeal==0] <- "z_Indifferent"
test_wine_imp1$LabelAppeal[test_wine_imp1$LabelAppeal==1] <- "Pos_One"
test_wine_imp1$LabelAppeal[test_wine_imp1$LabelAppeal==2] <- "Pos_Two"

test_wine_imp1$LabelAppeal <- as.factor(test_wine_imp1$LabelAppeal)

#Recode STARS variable; Turn NA's into "Missing"
test_wine_imp1$STARS <- as.character(test_wine_imp1$STARS)

test_wine_imp1$STARS[test_wine_imp1$STARS==1] <- "One_Star"
```

```
test_wine_imp1$STARS[test_wine_imp1$STARS==2] <- "Two_Stars"
test_wine_imp1$STARS[test_wine_imp1$STARS==3] <- "Three_Stars"
test_wine_imp1$STARS[test_wine_imp1$STARS==4] <- "Four_Stars"

test_wine_imp1$STARS <- as.factor(test_wine_imp1$STARS)

#Check percentile values and possibly do windsorizing
fa_high1 <- quantile(train_wine_imp1$FixedAcidity, c(.98))
va_high1 <- quantile(test_wine_imp1$VolatileAcidity, c(.98))
rs_high1 <- quantile(test_wine_imp1$ResidualSugar, c(.98))
ca_high1 <- quantile(test_wine_imp1$CitricAcid, c(.98))
c_high1 <- quantile(test_wine_imp1$Chlorides, c(.98))
fsd_high1 <- quantile(test_wine_imp1$FreeSulfurDioxide, c(.98))
tsd_high1 <- quantile(test_wine_imp1$TotalSulfurDioxide, c(.98))
d_high1 <- quantile(test_wine_imp1$Density, c(.98))
ph_high1 <- quantile(test_wine_imp1$pH, c(.98))
s_high1 <- quantile(test_wine_imp1$Sulphates, c(.98))
a_high1 <- quantile(test_wine_imp1$Alcohol, c(.98))
ai_high1 <- quantile(test_wine_imp1$AcidIndex, c(.98))

test_wine_imp1$FixedAcidity[test_wine_imp1$FixedAcidity > fa_high1] <- fa_high1
test_wine_imp1$VolatileAcidity[test_wine_imp1$VolatileAcidity > va_high1] <- va_high1
test_wine_imp1$ResidualSugar[test_wine_imp1$ResidualSugar>rs_high1] <- rs_high1
test_wine_imp1$CitricAcid[test_wine_imp1$CitricAcid>ca_high1] <- ca_high1
test_wine_imp1$Chlorides[test_wine_imp1$Chlorides>c_high1] <- c_high1
test_wine_imp1$FreeSulfurDioxide[test_wine_imp1$FreeSulfurDioxide>fsd_high1] <- fsd_high1
test_wine_imp1$TotalSulfurDioxide[test_wine_imp1$TotalSulfurDioxide>tsd_high1] <- tsd_high1
test_wine_imp1$pH[test_wine_imp1$pH>ph_high1] <- ph_high1
test_wine_imp1$Sulphates[test_wine_imp1$Sulphates>s_high1] <- s_high1
test_wine_imp1$Alcohol[test_wine_imp1$Alcohol>a_high1] <- a_high1
test_wine_imp1$AcidIndex[test_wine_imp1$AcidIndex>ai_high1] <- ai_high1

#Create new variables
test_wine_imp1$AcidicWine_Flag <- test_wine_imp1$AcidIndex
test_wine_imp1$AcidicWine_Flag <- ifelse(test_wine_imp1$AcidicWine_Flag >7, "z_AcidicWine",
"NonAcidicWine")
test_wine_imp1$AcidicWine_Flag <- as.factor(test_wine_imp1$AcidicWine_Flag)

test_wine_imp1$SaltyWine_Flag <- test_wine_imp1$Chlorides
test_wine_imp1$SaltyWine_Flag <- ifelse(test_wine_imp1$SaltyWine_Flag > 0.2, "z_SaltyWine",
"NonSaltyWine")
test_wine_imp1$SaltyWine_Flag <- as.factor(test_wine_imp1$SaltyWine_Flag)

test_wine_imp1$StrongWine_Flag <- test_wine_imp1$Alcohol
test_wine_imp1$StrongWine_Flag <- ifelse(test_wine_imp1$StrongWine_Flag > 15, "z_StrongWine",
"WeakWine")
test_wine_imp1$StrongWine_Flag <- as.factor(test_wine_imp1$StrongWine_Flag)
```

```
#Square Root Transformation
test_wine_imp1$VolatileAcidity <- sqrt(test_wine_imp1$VolatileAcidity)
test_wine_imp1$ResidualSugar <- sqrt(test_wine_imp1$ResidualSugar)
test_wine_imp1$CitricAcid <- sqrt(test_wine_imp1$CitricAcid)
test_wine_imp1$Chlorides <- sqrt(test_wine_imp1$Chlorides)
test_wine_imp1$FreeSulfurDioxide <- sqrt(test_wine_imp1$FreeSulfurDioxide)
test_wine_imp1$TotalSulfurDioxide <- sqrt(test_wine_imp1$TotalSulfurDioxide)



#Score test data and save predictions as P_TARGET
test_wine_imp1$P_TARGET <- predict(zip.glm, newdata = test_wine_imp1, type="response")

#Scored data file
scores <- test_wine_imp1[c("INDEX", "P_TARGET")]
write.csv(scores, file = "Scored_WineSales.csv", row.names = FALSE)
```