Kaggle Name: Christopher Jose

Introduction

The objective of this report is to build a model that predicts the number of wine cases ordered for a given wine, based upon characteristics of that wine. A dataset of 12,000 wines is used to build multiple models, from which one model is selected.

This report consists of the following steps, with the above goal in mind:
1. Exploratory Data Analysis (EDA)
2. Data Preparation
3. Model Building
4. Model Selection

I do EDA to know how to best prepare the data in the Data Preparation step. In the Model Building step, I go over eight models that I have built off data prepared in the Data Preparation step. I then compare the eight models in the Model Selection step to select the optimal model. The optimal model can then be used to predict the number of wine cases ordered for a given wine, assuming data for the variables used in the model is available for that wine.

The interesting aspect of this dataset is that the response variable is a count variable (a positive integer). There are several models that are well suited for count response variables: Poisson, Negative Binomial, Zero Inflated Poisson, and Zero Inflated Negative Binomial Regression. I consider these models, as well as Linear Regression, Decision Trees, Random Forests, and Gradient Boosted Regression Trees.

EXPLORATORY DATA ANALYSIS

I examine the data to answer the following questions:
1. Which models would work, given that we're predicting a count variable (positive integer)?
2. Are there missing values that need to be imputed?
3. Which predictors will likely be predictive of selling wine cases?
4. Are there outliers that need to be capped?

The dataset contains 12795 observations and 15 variables. All 14 variables will be considered in predicting the dependent variable, TARGET, which is a count variable representing the number of wine cases purchased by wine distribution companies. 11 of the 14 predictors are quantitative, continuous variables. 1 predictor is discrete: AcidIndex. 2 predictors are ordinal variables: LabelAppeal and STARS.

The data is as follows:

VARIABLE NAME	DEFINITION	THEORETICAL EFFECT
INDEX	Identification Variable (do not use)	None
TARGET	Number of Cases Purchased	None
AcidIndex	Proprietary method of testing total acidity of wine by using a weighted average	
Alcohol	Alcohol Content	
Chlorides	Chloride content of wine	
CitricAcid	Citric Acid Content	
Density	Density of Wine	
FixedAcidity	Fixed Acidity of Wine	
FreeSulfurDioxide	Sulfur Dioxide content of wine	
LabelAppeal	Marketing Score indicating the appeal of label design for consumers. High numbers suggest customers like the label design. Negative numbers suggest customes don't like the design.	Many consumers purchase based on the visual appeal of the wine label design. Higher numbers suggest better sales.
ResidualSugar	Residual Sugar of wine	
STARS	Wine rating by a team of experts. 4 Stars = Excellent, 1 Star = Poor	A high number of stars suggests high sales
Sulphates	Sulfate conten of wine	
TotalSulfurDioxide	Total Sulfur Dioxide of Wine	
VolatileAcidity	Volatile Acid content of wine	
pH	pH of wine	

Table 1: Data Dictionary

1. Which models deserve consideration?

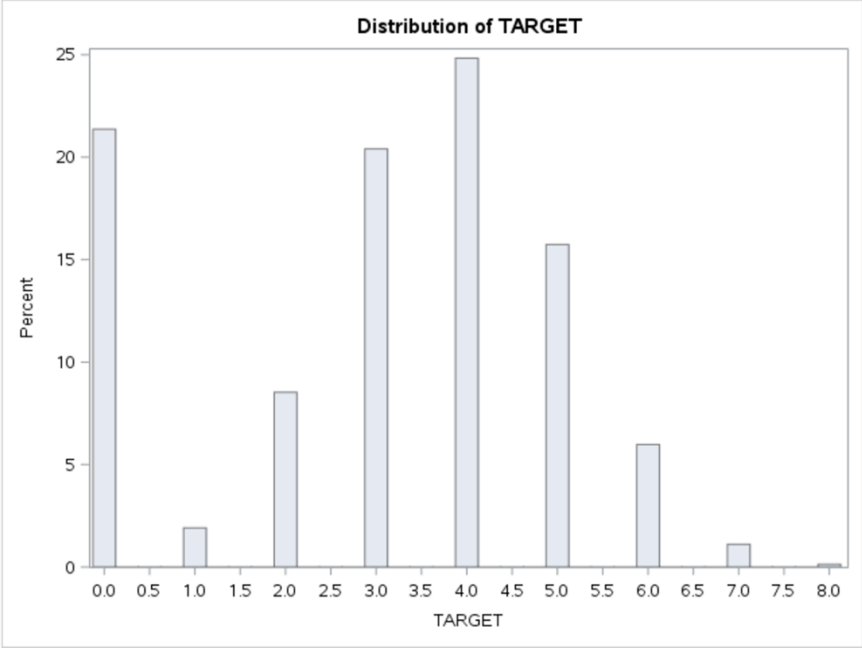
Poisson, Negative Binomial, Zero Inflated Poisson, Zero Inflated Negative Binomial, or Hurdle models are often candidates for predicting a count variable (positive integer), which we have here. I will also consider other models, just in case they perform better.

Firstly, if the count variable appears to have a right skewed distribution, then Poisson or Negative Binomial regression are good candidates. Also, Poisson regression assumes for the response variable that: **mean = variance**, so if we observe this to be the case in the data, then Poisson regression might be preferable. Also, Poisson assumes observations are independent. When this second assumption does not hold, it is often the case that mean < variance, a situation known as overdispersion. In this situation, Negative Binomial regression might be preferable, since it assumes for the response variable that: **mean < variance**.

If there are a large number of values equaling zero, then using a Zero Inflated Poisson Regression (ZIP Regression), Zero Inflated Negative Binomial Regression (ZINB Regression), or Hurdle model might be preferable.

With these thoughts in mind, let's inspect the response variable's distribution, as shown below:

Analysis Variable : TARGET	
Mean	Variance
3.0290739	3.7108945



Histogram of Response Variable

The variance is greater than the mean, but not by much. Also, while the distribution appears normally distributed (except for the spike of zero values), Poisson approximates a normal distribution as its mean increases. Negative Binomial also approximates a normal distribution as the sample size increases. Assuming these two distributions behind the data is thus still reasonable. Since there is a spike of zero values, the ZIP, ZINB, or Hurdle models deserve consideration and might very well outperform the other models.

2. Are there missing values that need to be imputed?

There are missing values that need to be dealt with. 8 predictors have missing values, with STARS having 5 times the number of missing values as most variables with missing values.

Variable	N Miss
INDEX	0
TARGET	0
FixedAcidity	0
VolatileAcidity	0
CitricAcid	0
ResidualSugar	616
Chlorides	638
FreeSulfurDioxide	647
TotalSulfurDioxide	682
Density	0
pH	395
Sulphates	1210
Alcohol	653
LabelAppeal	0
AcidIndex	0
STARS	3359

Missing Value Frequency Count

I examine missing STARS values further. As shown below, missing STARS values appear to be negatively correlated with TARGET. Most missing values occur for TARGET=0, when the number of wine cases bought is zero. Wine distribution companies appear to buy less wine for wines that have not had a wine rating entered into the system. This suggests that a flag variable indicating whether or not the STARS value is missing will be predictive of TARGET.

	TARGET									
STARS	0	1	2	3	4	5	6	7	8	Total
.	2038	126	335	457	260	101	32	8	2	3359

Missing STARS Frequency Count by TARGET value

3. Which predictors will likely be predictive of selling wine cases

I examine which predictors will likely be the most predictive of TARGET by looking at Pearson Correlation Coefficients. A predictor with a higher coefficient likely indicates greater predictability of TARGET in the models. While there are ordinal and nominal variables (predictors indicating missing variables), these coefficients offer a rough look at predictability.

Predictors in order of decreasing importance, as measured by the absolute value of their Pearson Correlation Coefficients (flag variables indicating missing values are included):

Predictor	Pearson Correlation Coefficient
m_STARS	-0.57
STARS	0.56
LabelAppeal	0.36
AcidIndex	-0.25

m_STARS is a flag variable indicating whether a STARS value is missing. M_STARS has a moderate, negative correlation of -.57, indicating that TARGET tends to drop when the STARS value is missing.

Most predictors have a very weak correlation with TARGET and with each other, so multicollinearity should not be an issue:

	TARGET	FixedAcidity	VolatileAcidity	CitricAcid	ResidualSugar	Chlorides	FreeSulfurDioxide	TotalSulfurDioxide	Density	pH	Sulphates	Alcohol	LabelAppeal	AcidIndex	STARS	m_ResidualSugar	m_Chlorides	m_FreeSulfurDioxide	m_TotalSulfurDioxide	m_pH	m_Sulphates	m_Alcohol	m_STARS
TARGET	1.00																						
FixedAcidity	-0.05	1.00																					
VolatileAcidity	-0.09	0.01	1.00																				
CitricAcid	0.01	0.01	-0.02	1.00																			
ResidualSugar	0.02	-0.02	-0.01	-0.01	1.00																		
Chlorides	-0.04	0.00	0.00	-0.01	-0.01	1.00																	
FreeSulfurDioxide	0.04	0.00	-0.01	0.01	0.02	-0.02	1.00																
TotalSulfurDioxide	0.05	-0.02	-0.02	0.01	0.02	-0.01	0.01	1.00															
Density	-0.04	0.01	0.01	-0.01	0.00	0.02	0.00	0.01	1.00														
pH	-0.01	-0.01	0.01	-0.01	0.01	-0.02	0.01	0.00	0.01	1.00													
Sulphates	-0.04	0.03	0.00	-0.01	-0.01	0.00	0.01	-0.01	-0.01	0.01	1.00												
Alcohol	0.06	-0.01	0.00	0.02	-0.02	-0.02	-0.02	-0.02	-0.01	-0.01	0.00	1.00											
LabelAppeal	0.36	0.00	-0.02	0.01	0.00	0.01	0.01	-0.01	-0.01	0.00	0.00	0.00	1.00										
AcidIndex	-0.25	0.18	0.04	0.07	-0.01	0.03	-0.04	-0.05	0.04	-0.06	0.03	-0.04	0.02	1.00									
STARS	0.56	-0.01	-0.03	0.00	0.02	0.00	-0.01	0.01	-0.02	0.00	-0.01	0.07	0.33	-0.09	1.00								
m_ResidualSugar	0.01	0.00	0.01	0.00		0.00	0.01	0.01	0.01	0.01	0.01	0.00	0.00	-0.01	0.01	1.00							
m_Chlorides	0.00	0.00	-0.01	-0.01	-0.02		0.00	0.00	0.00	0.00	0.01	0.01	0.00	-0.01	0.00	0.02	1.00						
m_FreeSulfurDioxide	0.00	0.01	0.00	0.00	0.00	0.01		0.01	0.01	0.00	0.00	0.01	-0.01	0.00	0.01	0.00	0.00	1.00					
m_TotalSulfurDioxide	0.01	0.00	-0.02	0.00	-0.02	-0.01	0.00		0.02	0.01	0.01	0.00	0.00	-0.01	0.01	0.00	0.00	0.02	1.00				
m_pH	-0.01	0.00	-0.02	0.01	0.01	0.00	0.00	0.01	-0.01		-0.01	0.01	0.00	0.00	0.00	0.00	-0.02	0.00	0.00	1.00			
m_Sulphates	-0.01	0.00	0.01	0.01	0.01	0.00	0.00	0.00	0.00	0.01		0.00	0.00	-0.01	-0.01	0.00	-0.02	-0.01	0.01	0.00	1.00		
m_Alcohol	0.00	0.00	0.00	-0.01	0.00	0.00	0.00	-0.01	0.00	0.01	0.01		-0.01	0.01	0.00	0.01	0.00	0.00	0.00	0.01	-0.01	1.00	
m_STARS	-0.57	0.04	0.06	-0.01	-0.01	0.03	-0.03	-0.03	0.02	0.01	0.03	-0.03	-0.11	0.17		0.00	0.00	0.02	0.01	0.01	0.01	0.01	1.00

Table 5: Pearson Correlation Coefficients

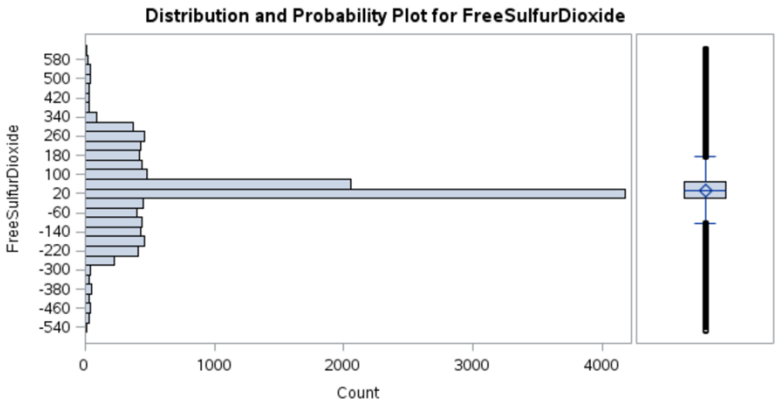
4. Are there outliers that need to be capped?

The range (max-min) for each predictor is small, except for TotalSulfurDioxide, FreeSulfurDioxide, and ResidualSugar. Mean and medians are close, indicating a lack of outliers.

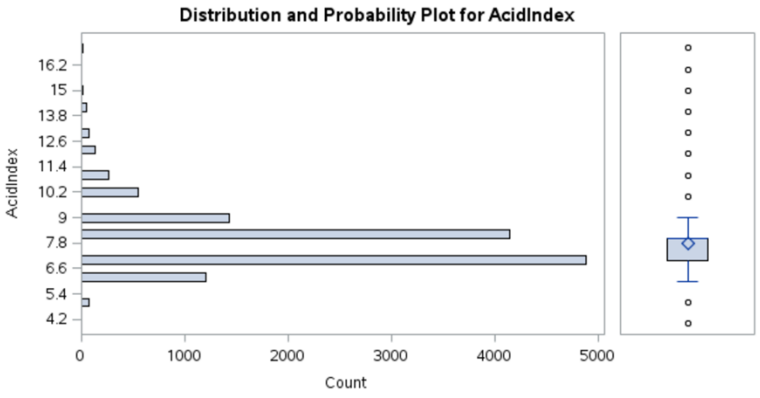
Variable	Mean	Median	Range
TARGET	3	3	8
FixedAcidity	7	7	53
VolatileAcidity	0	0	6
CitricAcid	0	0	7
ResidualSugar	5	4	269
Chlorides	0	0	3
FreeSulfurDioxide	31	30	1178
TotalSulfurDioxide	121	123	1880
Density	1	1	0
pH	3	3	6
Sulphates	1	1	7
Alcohol	10	10	31
LabelAppeal	-0	0	4
AcidIndex	8	8	13
STARS	2	2	3

All 11 quantitative, continuous variables appear normally distributed with long tails on either side. Discrete predictor, AcidIndex, appears right-skewed. Ordinal predictors, LabelAppeal and STARS, have small ranges and no outliers. Some data values are negative, though I do not have the knowledge to assess the reasonability of this fact. It appears that trimming the continuous and discrete variables might offer a better fit across our models, to deal with the extreme observations (outliers) in the continuous variables' wide ranges and discrete variable's right skewed discrete distribution.

Most distributions look like this:



AcidIndex's right-skewed distribution:



DATA PREPARATION

After exploring the data, I do the following:
1. Impute missing values of each numeric variable with its median.
2. Trim all non-ordinal predictors to reduce the impact of outliers on the model.
3. Consider cube root transforming certain variables

1. Impute missing values with the median

I "fill in" missing values with each variable's median, though since means are close to medians, the mean could have been used as well. Recall that the following variables have missing values: STARS, Sulphates, Alcohol, pH, TotalSulfurDioxide, FreeSulfurDioxide, Chlorides, ResidualSugar.

Flag variables are made for every variable and are labeled with the original variable's name, but with an M in front, where a 1 for an observation indicates that it had a missing value which was median imputed for the variable associated with the flag variable. As shown below, all variables have zero missing values (where *N Miss* indicates the number of missing values). Imputed values are shown in the column labeled "Median".

Variable	N Miss
INDEX	0
M_AcidIndex	0
acidindex	0
M_Alcohol	0
alcohol	0
M_Chlorides	0
chlorides	0
M_CitricAcid	0
citricacid	0
M_Density	0
density	0
M_FixedAcidity	0
fixedacidity	0
M_FreeSulfurDioxide	0
freesulfurdioxide	0

Variable	N Miss
M_LabelAppeal	0
labelappeal	0
M_ResidualSugar	0
residualsugar	0
M_STARS	0
stars	0
M_Sulphates	0
sulphates	0
M_TotalSulfurDioxide	0
totalsulfurdioxide	0
M_VolatileAcidity	0
volatileacidity	0
M_pH	0
pH	0

Variable	N Miss	Median
INDEX	0	8110.0
TARGET	0	3.0
FixedAcidity	0	6.9
VolatileAcidity	0	0.3
CitricAcid	0	0.3
ResidualSugar	616	3.9
Chlorides	638	0.0
FreeSulfurDioxide	647	30.0
TotalSulfurDioxide	682	123.0
Density	0	1.0
pH	395	3.2
Sulphates	1210	0.5
Alcohol	653	10.4
LabelAppeal	0	0.0
AcidIndex	0	8.0
STARS	3359	2.0

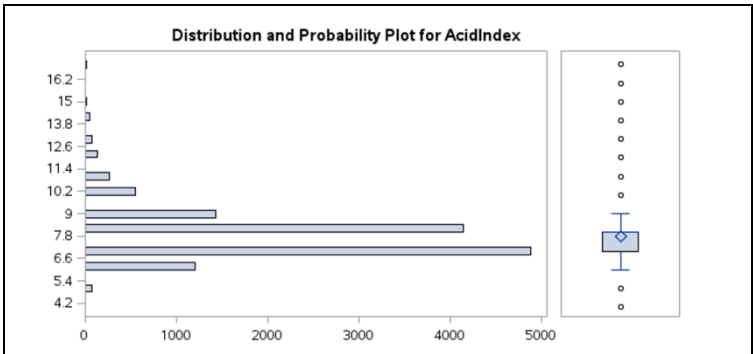
Since STARS is positively correlated with TARGET, it would make sense to impute STARS based on the value of TARGET in constructing the models. However, when scoring new data, and in the process, imputing missing STARS values in the new data, you can't impute missing STARS based on TARGET, since there are no TARGET values available for the new data. TARGET is what you are predicting, after all. Thus you would have to do something like median imputation for STARS values for the new data, and then your new data would be fed into models built based on a different imputation method. This would make the new data incongruent with the data used to make the models, causing a mismatch.

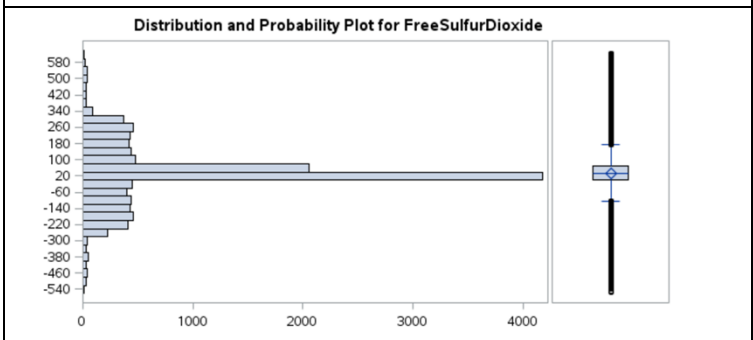
2. Trimming variables with large ranges and skewed distributions

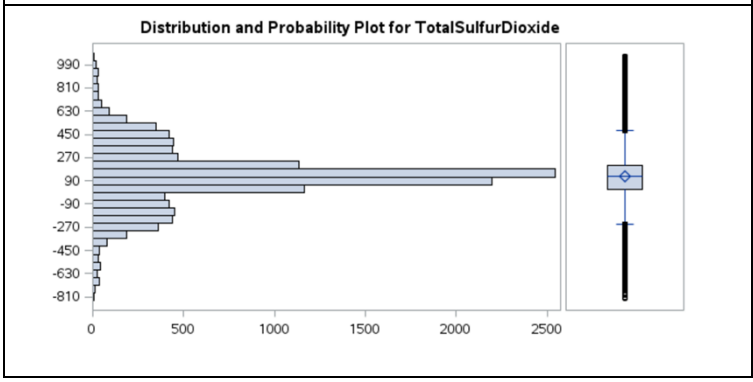
For all quantitative, continuous variables, as well as the discrete variable (AcidIndex), I've trimmed their data to stay within the range of the 1st and 99th percentiles (P1 and P99) of their original distributions.

Before and after displays of AcidIndex and the 3 continuous variables with the widest ranges are shown on the next page. The trimmed distributions for these predictors have smaller ranges, as expected.

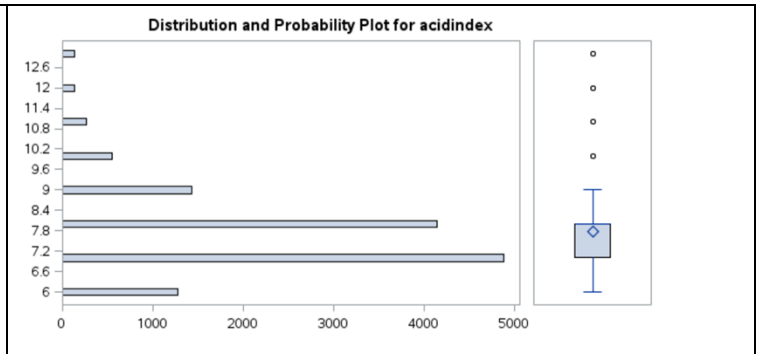
Before Trimming to P1 and P99

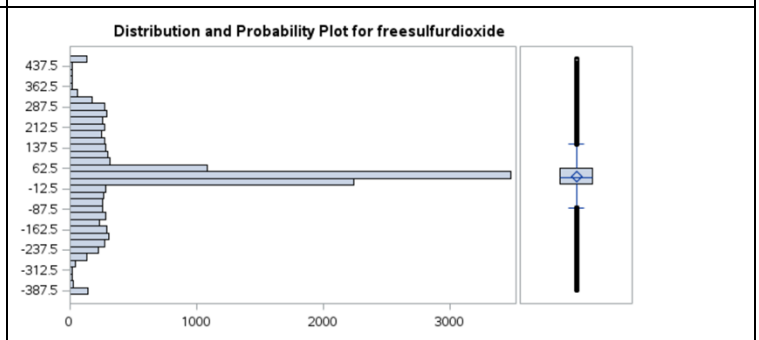


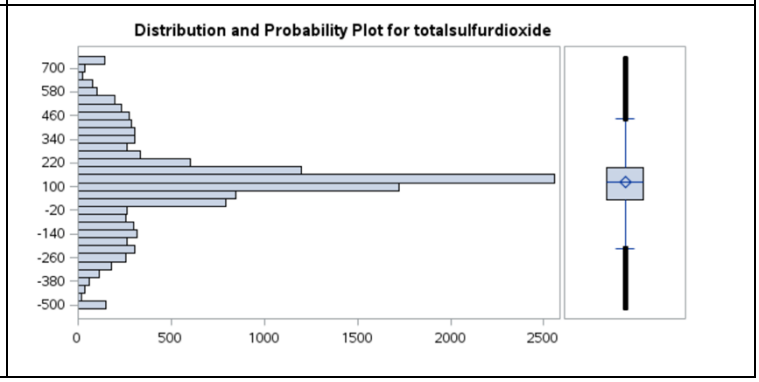


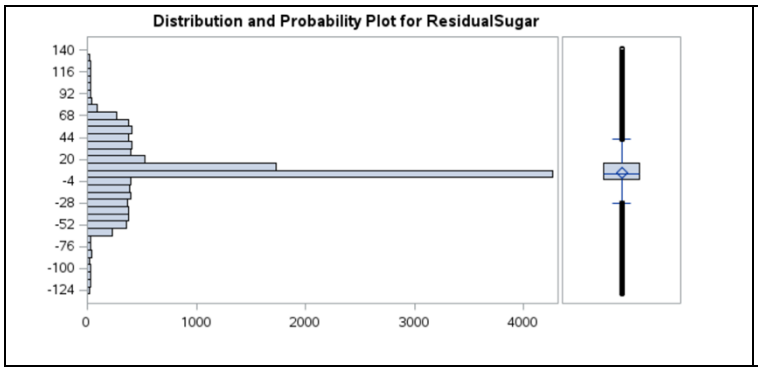


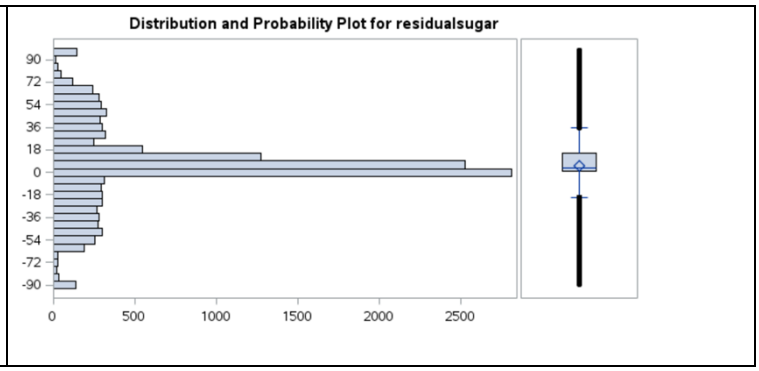
After Trimming to P1 and P99











3. Consider Cube Root Transforming Certain Variables

As shown on the previous page, the 3 continuous predictors with the widest ranges have had their distributions trimmed to the 1st and 99th percentiles. These predictors, ResidualSugar, TotalSulfurDioxide, and FreeSulfurDioxide, still have much wider ranges than all of the other variables. Considering that their Pearson correlation coefficients are very low (.05 or less), I consider doing a cube root transform of these predictors to dampen the potential negative impact of their wide distributions on the models. These predictors have negative values, so other transformations, like log or square root transform, would not have worked.

I find that there is almost no change in model results after cube root transforming these variables, so I leave them as is. See below for the lack of difference between regression results.

Using original variables

Linear Regression

Obs	_RMSE_	_ADJRSQ_	_AIC_	_BIC_
1	1.30777	0.53682	4828.67	4830.78

Poisson Regression

Criteria For Assessing Goodness Of Fit			
Criterion	DF	Value	Value/DF
Deviance	8935	9551.7310	1.0690
Scaled Deviance	8935	9551.7310	1.0690
Pearson Chi-Square	8935	7801.8568	0.8732
Scaled Pearson X2	8935	7801.8568	0.8732
Log Likelihood		6143.5209	
Full Log Likelihood		-15991.9773	
AIC (smaller is better)		32027.9546	
AICC (smaller is better)		32028.0679	
BIC (smaller is better)		32184.1588	

Using cube rooted variables

Linear Regression

Obs	_RMSE_	_ADJRSQ_	_AIC_	_BIC_
1	1.30777	0.53682	4828.67	4830.78

Poisson Regression

Criteria For Assessing Goodness Of Fit			
Criterion	DF	Value	Value/DF
Deviance	8935	9555.3220	1.0694
Scaled Deviance	8935	9555.3220	1.0694
Pearson Chi-Square	8935	7803.6413	0.8734
Scaled Pearson X2	8935	7803.6413	0.8734
Log Likelihood		6141.7254	
Full Log Likelihood		-15993.7728	
AIC (smaller is better)		32031.5456	
AICC (smaller is better)		32031.6589	
BIC (smaller is better)		32187.7498	

MODEL BUILDING

Model 1– Linear Regression

This model runs a linear regression model off all predictors, including predictors that have been median imputed and trimmed at the 1st and 99th percentiles, as well as flag variables created from median imputation.

Model 2– Poisson Regression

This model is a good choice for a count response variable and assumes the mean and variance are equal for TARGET, which they roughly are (variance is slightly above mean).

Model 3– Negative Binomial Regression

This model is a good choice for a count response variable and assumes the variance is greater than or equal to the mean for TARGET, which it is. This model uses only predictors that were significant at the 5% level in the Poisson regression.

Model 4– Zero Inflated Poisson (ZIP) Regression

This model accounts for the spike of zero values for the response variable. It runs a poisson regression to predict TARGET assuming it is >0. It runs a logistic regression to predict whether TARGET is not zero. Then it predicts TARGET by multiplying the predicted values from both regressions such that (TARGET | TARGET > 0) x P(TARGET is not zero) = Predicted TARGET for observation j.

Model 5– Zero Inflated Negative Binomial (ZINB) Regression

This model is similar to model 4, except it uses a negative binomial regression to predict (TARGET | TARGET > 0).

Model 6– Decision Trees

This model builds a regression tree and is included as a benchmark model.

Model 7– Random Forests

This model constructs a multitude of regression trees, typically several hundred. Each tree is different because of having been created from two randomization processes: tree bagging and feature bagging. Each tree spits out a prediction and the final prediction is based on the predictions from all of the trees.

Model 8– Gradient Boosted Trees

A gradient boosted regression tree starts out as just a regression tree that predicts y. Then, a regression tree that predicts the residuals of the model is created, and its residual predictions are added to each predicted y value, such that each yhat then equals

yhat1 = yhat + learning rate (predicted residual for yhat model)

Then another regression tree is created which models yhat1's residuals, such that each predicted value then equals

yhat2 = yhat1 + learning rate (predicted residual for yhat1 model).

This process is repeated for a predetermined number of boosting stages, meaning trees that improve the original tree.

Additional Comments

All models are built off a 70% train split and tested on the 30% test split, where their root mean squared errors are compared. Models 2-5 use the method of maximum likelihood to derive parameter estimates and so we can compare their AIC, AICC, BIC values. All models except model 3 use all predictors.

Model 1– Linear Regression

While the response variable is a count variable, making linear regression un-optimal, I implement it anyway, in part, as a benchmark model to compare with other models.

The model's adjusted r-squared is .5368, meaning 53.68% of the variability in target is accounted for by the model (by the regression line). This suggests the model offers fairly precise predictability of target in the train data and suggests predictability of target in new data. 9 predictors are statistically significant at the 5% level, meaning they definitely appear to have a relationship with target (their regression coefficients differ from zero), and can therefore predict the change in target, holding all other variables constant.

I only have knowledge to check the reasonability of coefficients for two predictors: LabelAppeal, STARS, and M_STARS. LabelAppeal's coefficient is .47842, indicating that the more customers like the label design, the more wine that is predicted to sell. The coefficient for STARS is .77379, indicating that a one-unit increase in STARS is predicted to increase target by .774. The coefficient for M_STARS is -2.23, indicating that when the STARS value is missing, wine cases sold is predicted to drop by 2.23. This coefficient represents the change in the intercept and the whole regression line is shifted downwards. All of the coefficients for these variables makes sense.

Standardized estimates in the table allows us to compare coefficients and determine which predictors are the most important: m_STARS, STARS, LabelAppeal AcidIndex.

All VIFs are close to 1, indicating that there are no multicollinearity problems.

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Standardized Estimate	Variance Inflation
Intercept	1	4.08001	0.54999	7.42	<.0001	0	0
AcidIndex	1	-0.21361	0.01136	-18.81	<.0001	-0.14136	1.09207
ALCOHOL	1	0.01476	0.00400	3.69	0.0002	0.02663	1.00869
CHLORIDES	1	-0.12252	0.04576	-2.68	0.0074	-0.01929	1.00314
CitricAcid	1	0.02328	0.01678	1.39	0.1654	0.01002	1.00837
Density	1	-0.40885	0.53950	-0.76	0.4486	-0.00546	1.00486
FixedAcidity	1	-0.00114	0.00228	-0.50	0.6173	-0.00366	1.03879
FREESULFURDIOXIDE	1	0.00015639	0.00009850	1.59	0.1124	0.01145	1.00481
LabelAppeal	1	0.47842	0.01635	29.27	<.0001	0.22174	1.10977
RESIDUALSUGAR	1	-0.00005464	0.00043472	-0.13	0.9000	-0.00090557	1.00380
STARS	1	0.77379	0.01884	41.08	<.0001	0.31077	1.10681
SULPHATES	1	-0.03523	0.01619	-2.18	0.0296	-0.01568	1.00451
totalsulfurdioxide	1	0.00019934	0.00006326	3.15	0.0016	0.02272	1.00490
VolatileAcidity	1	-0.08997	0.01847	-4.87	<.0001	-0.03517	1.00740
PH	1	-0.03026	0.02138	-1.42	0.1571	-0.01021	1.00675
m_ALCOHOL	1	0.10035	0.06205	1.62	0.1059	0.01165	1.00264
m_CHLORIDES	1	-0.01383	0.06365	-0.22	0.8280	-0.00157	1.00404
m_FREESULFURDIOXIDE	1	0.07445	0.06312	1.18	0.2382	0.00849	1.00180
m_RESIDUALSUGAR	1	0.04980	0.06376	0.78	0.4348	0.00563	1.00321
m_STARS	1	-2.22760	0.03215	-69.29	<.0001	-0.51008	1.04800
m_SULPHATES	1	0.00734	0.04660	0.16	0.8748	0.00113	1.00241
m_totalsulfurdioxide	0	0	.	.	.	.	.
m_PH	1	-0.08513	0.08132	-1.05	0.2952	-0.00753	1.00165

Model 2– Poisson Regression

Poisson regression assumes the response variable's mean = variance, which we roughly observed to be the case in the EDA. Poisson regression outputs parameter estimates that are such that:
$\ln(y) = b0 + b1x1 + b2x2 + \ldots + bjxj$
$y = \exp(b0 + b1x1 + b2x2 + \ldots + bjxj)$
$\exp(bj) - 1$ represents the % increase in y for a one-unit increase in predictor j

9 predictors have statistically significant coefficients at the 5% level (one is slightly above 5%). Predictor coefficients make sense. For example, target is predicted to increase by exp(.1861) – 1 = 20.45% for a one-unit increase in STARS. Predictors m_STARS, STARS, and LabelAppeal have large coefficients, as expected.

Compared with the linear regression model, in this model, every coefficient has the same sign but smaller magnitude. This means positive numbers become less positive and negative numbers become less negative. The intercept decreases by 2.45 points. The largest coefficient change is for m_STARS, whose coefficient goes from -2.23 to -1.01.

Analysis Of Maximum Likelihood Parameter Estimates							
Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	1.6293	0.2422	1.1546	2.1039	45.27	<.0001
AcidIndex	1	-0.0833	0.0056	-0.0943	-0.0723	220.27	<.0001
ALCOHOL	1	0.0042	0.0018	0.0008	0.0077	5.78	0.0162
CHLORIDES	1	-0.0395	0.0201	-0.0789	0.0000	3.84	0.0500
CitricAcid	1	0.0074	0.0074	-0.0070	0.0218	1.00	0.3162
Density	1	-0.1132	0.2370	-0.5777	0.3512	0.23	0.6328
FixedAcidity	1	-0.0007	0.0010	-0.0027	0.0012	0.53	0.4678
FREESULFURDIOXIDE	1	0.0001	0.0000	-0.0000	0.0001	1.78	0.1826
LabelAppeal	1	0.1631	0.0073	0.1488	0.1775	493.64	<.0001
RESIDUALSUGAR	1	-0.0000	0.0002	-0.0004	0.0004	0.01	0.9209
STARS	1	0.1861	0.0073	0.1717	0.2005	641.59	<.0001
SULPHATES	1	-0.0135	0.0071	-0.0274	0.0005	3.58	0.0584
totalsulfurdioxide	1	0.0001	0.0000	0.0000	0.0001	7.17	0.0074
VolatileAcidity	1	-0.0296	0.0081	-0.0456	-0.0137	13.26	0.0003
PH	1	-0.0121	0.0094	-0.0306	0.0063	1.67	0.1963
m_ALCOHOL	1	0.0298	0.0272	-0.0235	0.0831	1.20	0.2733
m_CHLORIDES	1	-0.0114	0.0277	-0.0657	0.0430	0.17	0.6822
m_FREESULFURDIOXIDE	1	0.0250	0.0278	-0.0294	0.0795	0.81	0.3681
m_RESIDUALSUGAR	1	0.0139	0.0277	-0.0404	0.0682	0.25	0.6166
m_STARS	1	-1.0089	0.0201	-1.0484	-0.9694	2507.09	<.0001
m_SULPHATES	1	0.0002	0.0206	-0.0402	0.0406	0.00	0.9922
m_totalsulfurdioxide	0	0.0000	0.0000	0.0000	0.0000	.	.
m_PH	1	-0.0374	0.0366	-0.1091	0.0343	1.04	0.3070
Scale	0	1.0000	0.0000	1.0000	1.0000		

Model 3– Negative Binomial Regression

Negative Binomial regression (NB regression) assumes the response variable's variance is above the mean, which it is slightly. Like Poisson regression, model outputs are such that:
$\ln(y) = b0 + b1x1 + b2x2 + \ldots + bjxj$

Using the same predictors as the previous models, NB regression yields the same parameter estimates as Poisson, so I run the NB regression with insignificant predictors removed at the 5% level (thus, I keep only the 9 predictors that were significant in the Poisson regression).

The dispersion parameter of zero indicates a lack of over-dispersion in the response variable, meaning the variance is not much above the mean. Thus, a Poisson regression is appropriate, though we can still use NB regression.

The coefficients on all predictors remain significant at the 5% level and change very little from the Poisson regression. The only noticeable change is in the intercept, which decreases by .15.

Parameter	DF	Estimate	Pr > ChiSq
Intercept	1	1.4791	<.0001
AcidIndex	1	-0.0834	<.0001
ALCOHOL	1	0.0042	0.0163
CHLORIDES	1	-0.0393	0.0510
LabelAppeal	1	0.1632	<.0001
STARS	1	0.1862	<.0001
SULPHATES	1	-0.0130	0.0662
totalsulfurdioxide	1	0.0001	0.0072
VolatileAcidity	1	-0.0297	0.0003
m_STARS	1	-1.0099	<.0001
Dispersion	0	0.0000	

I show the first five observations in the test split for both the Poisson and NB regressions. Notice that y_nb is always lower than y_poi, which makes sense because the NB regression's intercept is lower and this is the only difference between the two models.

y_poi	y_nb
3.5120236944	3.4619265085
3.3845943688	3.3823904311
1.2294662847	1.2249545801
5.1496478639	5.1094054162
2.9981027471	2.944322976

Model 4– Zero Inflated Poisson Regression

This model (ZIP regression) accounts for the spike of zero values in the response variable by generating two models, a logistic model and poisson model. Thus, two sets of parameter estimates are outputted. Model predictions are calculated as follows:

Poisson

$\ln(target\_amt \mid amt > 0) = b0 + b1x1 + b2x2 + \ldots + bjxj$

$target\_amt \mid amt > 0 = \exp(b0 + b1x1 + b2x2 + \ldots + bjxj)$

Logistic

$\ln(p / (1-p)) = c0 + c1x1 + c2x2 + \ldots + cjxj$

$p = 1 / (1+\exp(-(c0 + c1x1 + c2x2 + \ldots + cjxj)))$

where $p = P(target=0)$

Combined

predicted target = $(target\_amt \mid amt > 0) \times (1-p)$

Logistic Model

Parameter	DF	Estimate	Pr > ChiSq
Intercept	1	-1.7327	0.3048
AcidIndex	1	0.4674	<.0001
ALCOHOL	1	0.0316	0.0091
CHLORIDES	1	0.1301	0.3370
CitricAcid	1	-0.0505	0.3107
Density	1	-1.1722	0.4688
FixedAcidity	1	0.0056	0.4146
FREESULFURDIOXIDE	1	-0.0005	0.1044
LabelAppeal	1	0.7513	<.0001
RESIDUALSUGAR	1	-0.0009	0.4869
STARS	1	-3.6650	<.0001
SULPHATES	1	0.1822	0.0002
totalsulfurdioxide	1	-0.0011	<.0001
VolatileAcidity	1	0.2227	<.0001
PH	1	0.2272	0.0004
m_ALCOHOL	1	-0.2556	0.1659
m_CHLORIDES	1	0.1539	0.4373
m_FREESULFURDIOXIDE	1	-0.3095	0.1050
m_RESIDUALSUGAR	1	0.0961	0.6065
m_STARS	1	5.7779	<.0001
m_SULPHATES	1	0.0873	0.5180
m_totalsulfurdioxide	0	0.0000	.
m_PH	1	0.2136	0.3714

Poisson Model

Parameter	DF	Estimate	Pr > ChiSq
Intercept	1	1.3513	<.0001
AcidIndex	1	-0.0184	0.0021
ALCOHOL	1	0.0074	<.0001
CHLORIDES	1	-0.0206	0.3167
CitricAcid	1	0.0002	0.9800
Density	1	-0.2098	0.3892
FixedAcidity	1	-0.0000	0.9710
FREESULFURDIOXIDE	1	0.0000	0.7317
LabelAppeal	1	0.2378	<.0001
RESIDUALSUGAR	1	-0.0001	0.5488
STARS	1	0.1045	<.0001
SULPHATES	1	0.0001	0.9839
totalsulfurdioxide	1	-0.0000	0.2665
VolatileAcidity	1	-0.0112	0.1772
PH	1	0.0064	0.5040
m_ALCOHOL	1	0.0062	0.8242
m_CHLORIDES	1	-0.0063	0.8255
m_FREESULFURDIOXIDE	1	-0.0006	0.9826
m_RESIDUALSUGAR	1	0.0199	0.4835
m_STARS	1	-0.1814	<.0001
m_SULPHATES	1	0.0032	0.8795
m_totalsulfurdioxide	0	0.0000	.
m_PH	1	0.0001	0.9981
Scale	0	1.0000	

As shown in the tables above, 9 predictors are significant at the 5% level in the logistic model, while 5 are significant at the 5% level in the poisson model. Logistic parameter estimates are interpreted as: exp(cj)-1 is the percentage change in the odds for a one-unit increase in cj. For STARS, exp(-3.6650) – 1 = -.974, so there is a 97.4% decrease in the odds that target is zero for a one-unit increase in STARS. This makes sense because we observed that STARS is positively correlated with target, so as STARS increases, target increases. Poisson coefficients appear reasonable and are interpreted similarly to those in the Poisson regression model (model 2).

In comparing Poisson coefficients to those in the Poisson Regression (model 2), I observe changes in sign on coefficients for PH, m_PH, m_freesulfurdioxide, sulphates. The new values are very close to zero, however. Sign changes are logically possible because this model is for predicting target when target is greater than zero, while the Poisson regression (model 2) coefficients are for predicting target.

Model 5– Zero Inflated Negative Binomial Regression

ZINB regression is the similar to ZIP regression, except an NB model is used instead of a poisson model in predicting (target | target > 0). Model results are shown below:

For the logistic model, most coefficients change little from the ZIP model. All have the same sign. STARS, m_STARS, and the intercept have the largest changes:

STARS: -3.67 -> -1.89
m_STARS: 5.78 -> 4.04
intercept: -1.73 -> -3.23

For the negative binomial model, all coefficients change very little.

Logistic Model

Intercept	1	-3.2301
AcidIndex	1	0.4423
ALCOHOL	1	0.0275
CHLORIDES	1	0.1319
CitricAcid	1	-0.0594
Density	1	-1.1821
FixedAcidity	1	0.0056
FREESULFURDIOXIDE	1	-0.0005
LabelAppeal	1	0.6850
RESIDUALSUGAR	1	-0.0009
STARS	1	-1.8931
SULPHATES	1	0.1659
totalsulfurdioxide	1	-0.0010
VolatileAcidity	1	0.2080
PH	1	0.2092
m_ALCOHOL	1	-0.2701
m_CHLORIDES	1	0.1627
m_FREESULFURDIOXIDE	1	-0.2737
m_RESIDUALSUGAR	1	0.0782
m_STARS	1	4.0428
m_SULPHATES	1	0.0626
m_totalsulfurdioxide	0	0.0000
m_PH	1	0.2569

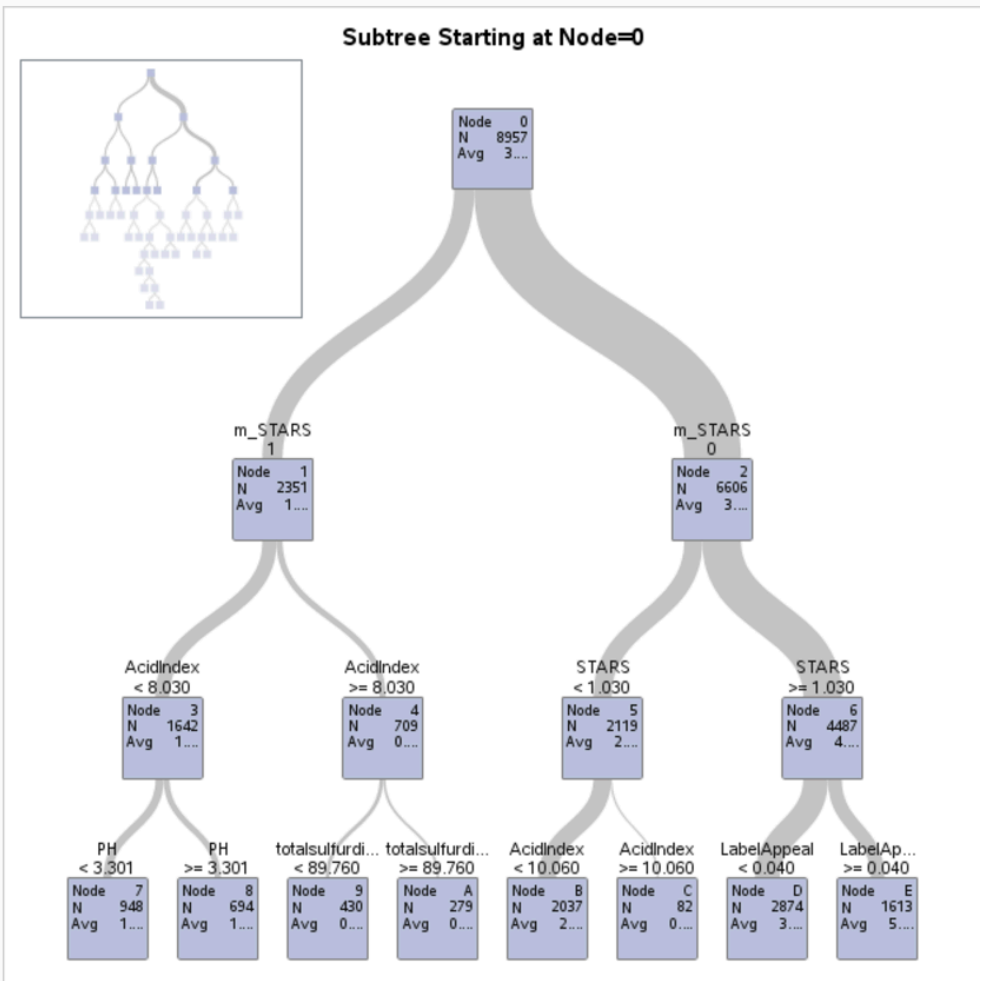
Negative Binomial

Parameter	DF	Estimate
Intercept	1	1.3493
AcidIndex	1	-0.0168
ALCOHOL	1	0.0073
CHLORIDES	1	-0.0195
CitricAcid	1	-0.0005
Density	1	-0.2162
FixedAcidity	1	0.0000
FREESULFURDIOXIDE	1	0.0000
LabelAppeal	1	0.2377
RESIDUALSUGAR	1	-0.0001
STARS	1	0.1040
SULPHATES	1	0.0005
totalsulfurdioxide	1	-0.0000
VolatileAcidity	1	-0.0113
PH	1	0.0068
m_ALCOHOL	1	0.0031
m_CHLORIDES	1	-0.0031
m_FREESULFURDIOXIDE	1	0.0006
m_RESIDUALSUGAR	1	0.0179
m_STARS	1	-0.1793
m_SULPHATES	1	0.0017
m_totalsulfurdioxide	0	0.0000
m_PH	1	0.0050
Dispersion	0	0.0019

Model 6– Decision Trees

This method builds a regression tree. The 3 most important or predictive variables in the tree are m_STARS, STARS, and LabelAppeal (as shown below). If an observation is dropped down the tree, then its m_STARS value is checked first. If m_STARS is zero, then the STARS value is inspected. If m_STARS is 1 then the AcidIndex value is inspected. This process continues until it hits the terminal nodes, where each node results in its own predicted value for target.

Variable Importance			
	Training		
Variable	Relative	Importance	Count
m_STARS	1.0000	103.4	1
STARS	0.6196	64.0670	4
LabelAppeal	0.5359	55.4107	5
AcidIndex	0.2399	24.8094	3
VolatileAcidity	0.1536	15.8831	3
totalsulfurdioxide	0.0878	9.0802	3
PH	0.0865	8.9441	1
ALCOHOL	0.0712	7.3620	2
FixedAcidity	0.0516	5.3362	1
SULPHATES	0.0466	4.8140	1



Model 7– Random Forests

I implement a Random Forest (RF) regression model with 300 regression trees in Python, where I tune the *max_features* parameter using 5-fold cross-validation and grid search. I consider 4 values in the grid search: sqrt(total features), 20%, 50%, and 80% of total features.

max_features sets the size of the random subset of features generated at each split in every tree, so all trees and splits will have the same number of randomly selected features. This parameter affects the feature bagging randomization step in constructing trees. A higher value can lead to over-fitting and less generalizability of the model, though too low of a value can also negatively affect model performance.

The procedure works as follows: the training data is split into 5 folds. Each fold is treated as a validation fold from which to evaluate a model that is trained on the other 4 folds. Thus, five models are created for each max_features value (one model for each fold), and 5 mean squared errors (MSEs') are averaged from having tested each model on its respective validation fold.

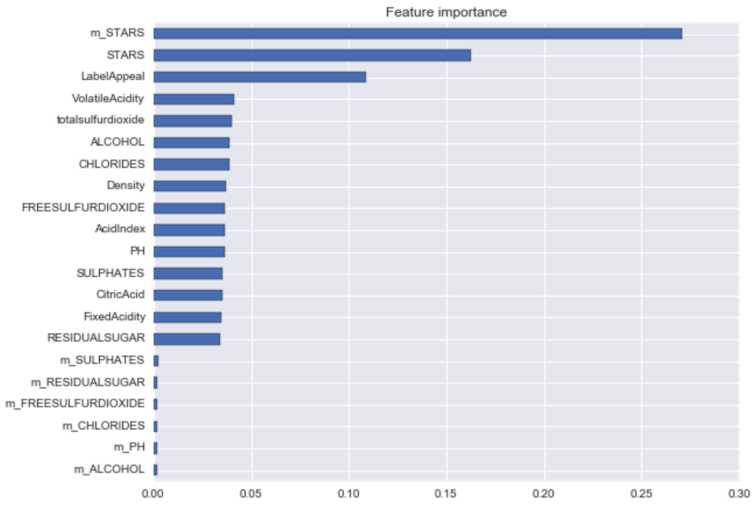
The algorithm I use, *GridSearchCV*, actually calculates the negative of the mean squared error (neg MSE) for each considered max_features value, so it selects the max_features value that maximizes the neg MSE.

The algorithm chooses a max_features value of .5, which produces the lowest neg MSE of -1.56. I then fit the random forest with the tuned max_features value of .5 on the train split.

The most important predictors in the random forest are m_STARS, STARS, and LabelAppeal.

```
from sklearn.model_selection import GridSearchCV
RANDOM_STATE=5
clf=RandomForestRegressor(warm_start=True, oob_score=True,random_state=RANDOM_STATE,
                          n_estimators=300)
parameters={"max_features":['sqrt',.2,.5,.8]}
fitmodel = GridSearchCV(clf, param_grid=parameters, cv=5, scoring="neg_mean_squared_error")
fitmodel.fit(X_train, Y_train)
fitmodel.best_estimator_, fitmodel.best_params_, fitmodel.best_score_, fitmodel.cv_results_
```

```
(RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=None,
            max_features=0.5, max_leaf_nodes=None, min_impurity_split=1e-07,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=300, n_jobs=1,
            oob_score=True, random_state=5, verbose=0, warm_start=True),
 {'max_features': 0.5},
 -1.5633955478644883,
```



Model 8– Gradient Boosted Trees

I implement a gradient boosted regression tree (GBRT) in Python, where I tune the *learning_rate* using 5-fold cross-validation and grid search.

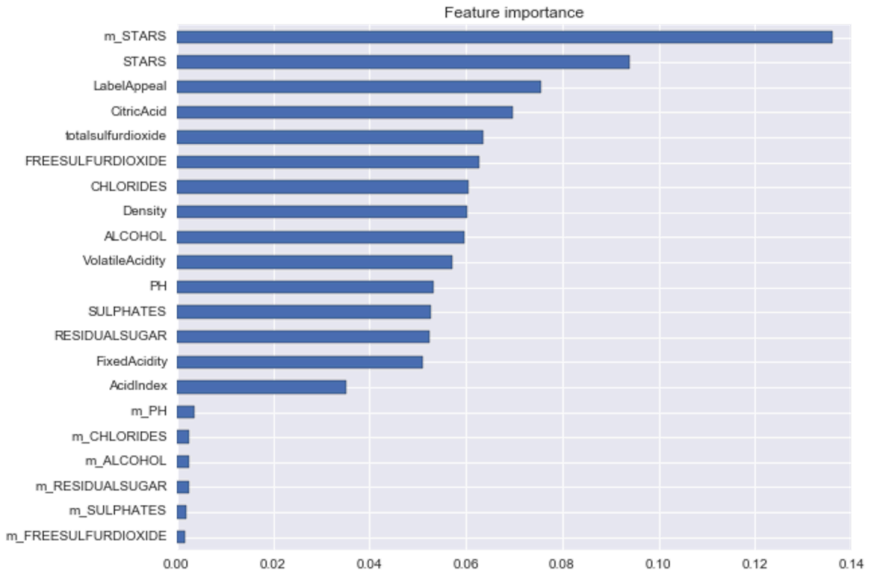
learning_rate controls the impact of each subsequent tree on the current model. A lower learning rate can control and lesson the impact that each tree has on the overall model, and thus prevent over-fitting.

The values that I use for other parameters (as shown in the code below) are within recommended ranges provided by the creator of XGBoost, Tianqi Chen. Grid search selects a *learning_rate* of .02, which I then use in fitting the final model.

The most important variables, as shown below, are m_STARS, STARS, and LabelAppeal.

```
parameters={'min_samples_split': 2,'n_estimators':300,'subsample':.75,
            'random_state':RANDOM_STATE, 'max_features':.6, 'max_depth':6}
clf = ensemble.GradientBoostingRegressor(**parameters)
parameters_grid={'learning_rate':[x/300.0 for x in range(2,11)]}
fitmodel = GridSearchCV(clf, param_grid=parameters_grid, cv=5, scoring="neg_mean_squared_error")
fitmodel.fit(X_train, Y_train)
fitmodel.best_estimator_, fitmodel.best_params_, fitmodel.best_score_, fitmodel.cv_results_
```

```
(GradientBoostingRegressor(alpha=0.9, criterion='friedman_mse', init=None,
              learning_rate=0.02, loss='ls', max_depth=6, max_features=0.6,
              max_leaf_nodes=None, min_impurity_split=1e-07,
              min_samples_leaf=1, min_samples_split=2,
              min_weight_fraction_leaf=0.0, n_estimators=300,
              presort='auto', random_state=5, subsample=0.75, verbose=0,
              warm_start=False),
 {'learning_rate': 0.02},
 -1.5242187616544787,
```



MODEL SELECTION

I select models based on their root mean squared error (rmse), which is associated with having evaluated them on the 30% test split. If two models are close in rmse, then if they have AIC, AICC, or BIC criterion, I will then select the model with the lower of these criterion. For all of these criterion, the smaller the value, the better.

	Linear	Poisson	NB	ZIP	ZINB	Decision Trees	Random Forests	GBRT
rmse (minimize)	1.314	1.321	1.323	1.340	1.340	1.290	1.256	**1.245**
AIC (minimize)		32028	32015	**28625**	28734			
AICC (minimize)		32028	32015	**28626**	28735			
BIC (minimize)		32184	32093	**28938**	29054			

I select the Gradient Boosted Regression Tree model because it has the lowest rmse. While this model is not as interpretable and easy to explain as the regression models, it offers a sizable reduction in rmse. Among the regression models, ZIP regression performed the best in terms of having the lowest AIC, AICC, and BIC criterion. However, it was the worst model (along with ZINB) in terms of rmse. Surprisingly, the linear regression model actually had the lowest rmse among regression models. I created a scoring program for the ZIP regression model (to score new data).

CONCLUSION

Gradient boosted regression trees have become very popular because of their ability to handle different kinds of data and still spit out a precise answer, as was shown to be the case here. The response variable was a count variable with a spike of zero values, and predictors ranged from continuous, discrete, and ordinal. In general, when the response is a count variable, Poisson and NB regression deserve strong consideration, as opposed to just using linear regression. When the response is a count variable with a spike of zero values, as was the case for this dataset, ZIP, ZINB or a hurdle model also deserve strong consideration. While the different regression models did not perform as well as Decision Trees, Random Forests, or GBRT, they are much more interpretable and still produce viable answers.

Regarding wine selling, I would suggest focusing on wine that has higher STARS and LabelAppeal values, meaning higher wine ratings and marketing scores. These variables were the most predictive of wine cases purchased in every model.