Assignment #3 - MSDS 411 Summer 2018

Andrew Knight

Introduction

My complete work for this assignment can be found in the Jupyter Notebook file **411_Assignment_3_Andrew_Knight.ipynb**.

The goal of this assignment is to predict the number of cases of wine sold using attributes of the wine as input. To do this, build at least three different models using 1. Negative Binomial Distribution 2. Poisson Distribution and 3. Regression. Optionally try the Hurdle model as well.

Define the metric for selecting the 'best' model and choose the final model to use. The scored data file containing the final model predictions on the test set will be submitted and scored on Kaggle.

Data Exploration (EDA)

The data contains specific attributes of wines. There are 14 independent variables for both train and test sets that describe these attributes for each wine record. All variables are given as numerics however it is clear from a quick glance that some scaling (regularization) of the variables may be required.

The data also contains several null values, listed as NaN. The variables in both train and test which have at least one null value are **residualsugar, chlorides, freesulfurdioxide, totalsulfurdioxide, ph, sulphates, alcohol** and **stars**. Another interesting thing about the null data is that the **stars** variable data consists of roughly 25% null values in both train and test. Due to such a high percentage of null data representation, it may be unwise to try to impute these values with mean or median during the cleaning step. Instead I may choose to simply leave this variable out of my model.

Finally, I noticed some negative values which will need a bit more research in the possible range to determine if these values are erroneous and need to be imputed. I also changed the variable (column) names to all lower case for consistency before proceeding with exploratory data analysis.

My observations specific to the training and test sets are given below.

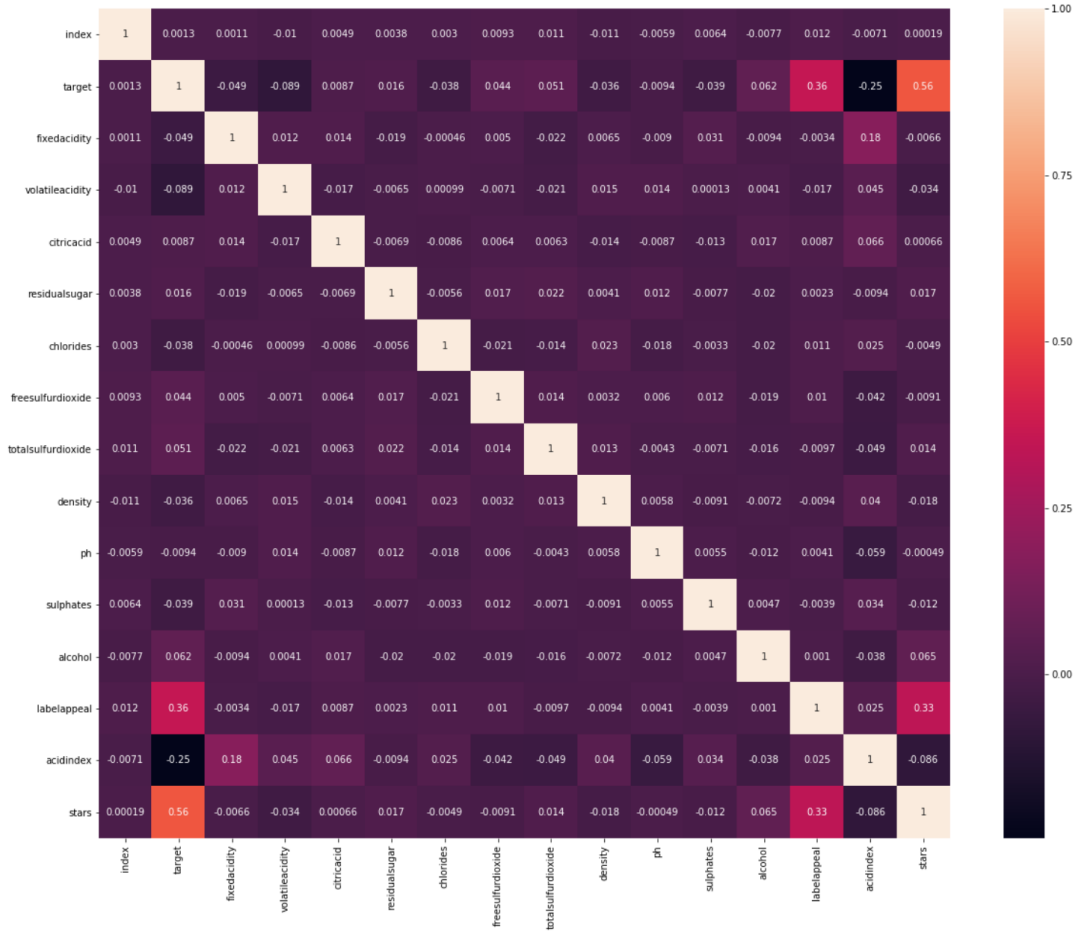
The Training Data:
There are 12,795 observations and 14 independent variables in the training data set. All data is given as numerics, some as ints and some as floats.

The Test Data:
The test data has 3,335 observations and 14 variables just as the training set. The response variable target is obviously null in this test set. All data is given as numerics, some as ints and some as floats.

Additional Observations from EDA:
It's worth noting that the majority of these data provide metrics about the composition and characteristics about the actual wine. However, two variables in the data set measure human sentiment rather than physical characteristics of the wine. This is an important distinction to consider for our model building because there is a level of subjectivity present and we must have confidence in the sample population used to determine the **labelappeal** and the **stars** values.

Some additional data that could be helpful if we were to try to expand our analysis and data gathering efforts include data about the location and or producer of the wine, year and month the wine was produced or bottled, and perhaps even the years of experience in wine production. The reason I think this would be useful is that we may be able to use this information along with subjective evaluation metrics **labelappeal** and **stars** variables to perhaps build a separate model based on consumer feedback to compare to a model for the physical attributes of the wine. In the end, we have to be mindful of the fact that we are trying to predict the amount of wine sold and our model is primarily built on measurable taste characteristics. Many more factors go into buyer decisions than just look, smell, and taste of the wine itself. For this preliminary analysis we must restrict our thinking to exclude these other factors like market dynamics, buyer influences, distribution channels and the like.

I also decided to try a heat map for viewing correlations between variables in the training data. Using the corr( ) function the plot is shown below.



The first thing I noticed about this plot is that the human sentiment items **labelappeal** and **stars** seem correlated as expected. The second thing that was evident from this is that these two variables also show the highest level of individual correlation to the target dependent variable. Based on this I may not want to completely omit the stars variable from my model if my ultimate goal is accuracy; I will need to test further by tuning the X parameters during my model building phase below.

Data Preparation

First, I needed to address the NaN values present in the train and test datasets. To do this, I chose to simply replace the NaN values with the mean for variables **residualsugar, chlorides, freesulfurdioxide, totalsulfurdioxide, ph, sulphates,** and **alcohol.** For the final variable containing null values, **stars** I first tried using the median value to replace the nulls.

Second, I needed to address the negative values occurring in several of the independent variables. For Negative Binomial and Poisson models, I want to be working with positive dependent variable data. One way to handle this is to apply a uniform shift to each variable by taking the absolute value of the minimum plus one and adding it to each record value. This ensures the new minimum value will be greater than zero and that the shift does not affect the correlations between values of a given variable.

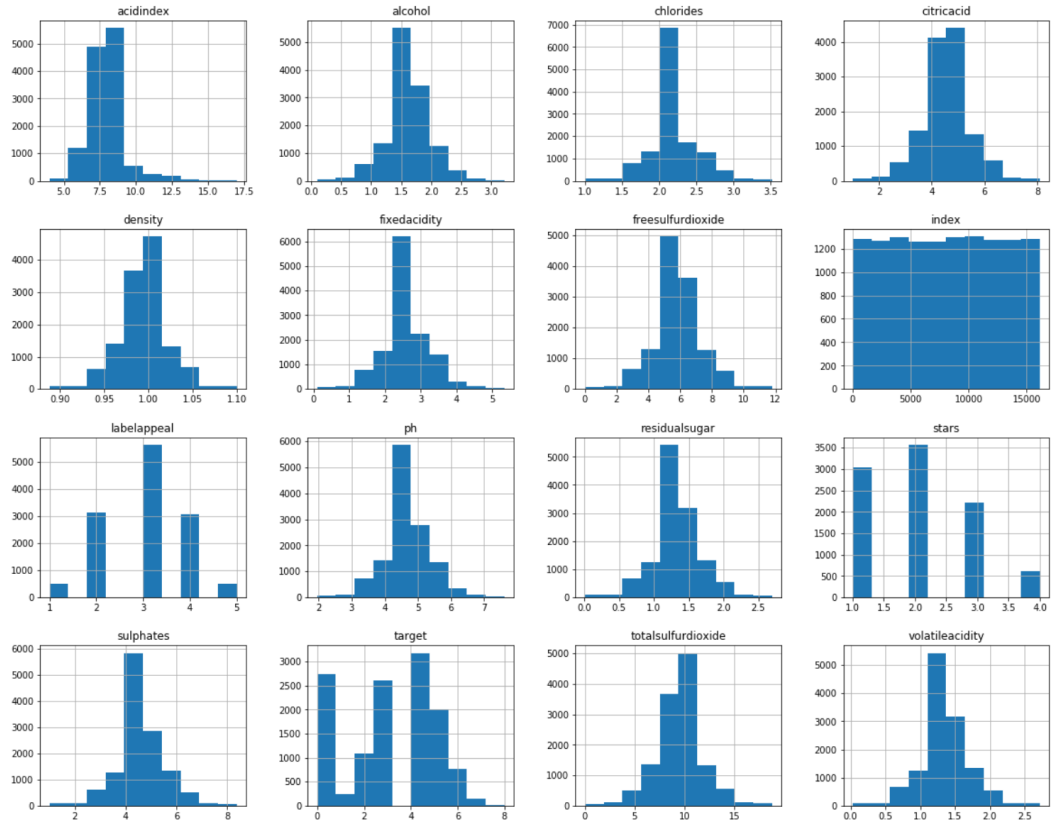
Third, I began creating a couple new variables to try using however I decided that without a good understanding of how each of the components affected wine tastes (clearly outside my area of domain

knowledge) it was difficult to create meaningful combinations of existing variables. I also decided based on the histogram plots that there would probably be little gain from creating log- or sqrt-transformed variables. Thus, no dummy or transformed variables were added to the train or test datasets. The fact that we started with all numeric variables also helped the cause.

Next, for the two variables **labelappeal** and **acidindex** which were originally defined as integers, I converted to float values for consistency in my numeric data for both train and test sets.

Finally, because I wanted to try several classification models and some of these do not handle unscaled data well (like SVC and SGD), I performed some basic scaling so that all variables are in the same general range. For variables with much larger relative values like **volatileacidity**, **freesulfurdioxide**, **totalsulfurdioxide**, and **residualsugar**, I chose to divide all values by 100. For the two variables with medium high relative values including **fixedacidity** and **alcohol**, I chose to divide all values by 10. This was performed on both train and test data.

After all of these steps were taken to clean the data, the plots were done again. The histograms are shown below for the training data. In general, these distributions look good and ready to be used in a variety of models. There are a couple minor adjustments I could make, for example the acidindex could possibly benefit from a log-transformation and it's still unclear to me how attributes such as **freesulfurdioxide** and **totalsulfurdioxide** are not highly correlated. However, I've decide to move forward with the model selection process using this cleaned data. These data were copied to new train and test data frames named **train1** and **test1**.



Build Models

I took a two-step approach to my building process that involved first selecting the variables I wanted to use for my different model trials and second testing the best set of variables on several different regression and classification models. This was don to try to limit the number of changing parameters when comparing different models. By starting with only three model options and testing my possible predictor variables (p1, p2, p3, p4) on only those three I should be able to determine the best set of predictors to use on a broader set of model choices.

The predictor variables I tested are listed below. I started with a simple set in p1 that only included sentiment variables. As expected this was far too simplistic however I found it interesting the values obtained between p1 and p3 were actually quite similar. This showed that while p2 listed several additional variables, the omission of **labelappeal** and **stars** significantly hurt the model performance. The **labelappeal** predictor was added back into **p3** and **stars** was also added in to **p4** along with other variables.

```
p0 = ["residualsugar", "chlorides", "freesulfurdioxide", "totalsulfurdioxide", "alcohol"] # trial 0 - shell code

p1 = ["labelappeal", "stars"] # trial 1

p2 = ["freesulfurdioxide", "totalsulfurdioxide", "alcohol", "citricacid", "density",
      "acidindex", "residualsugar"] # trial 2

p3 = ["chlorides", "freesulfurdioxide", "totalsulfurdioxide", "alcohol", "sulphates", "ph", "density",
      "acidindex", "citricacid", "labelappeal", "volatileacidity", "residualsugar"] # trial 3 - removed fixedacidity (not signif), not incl stars

p4 = ["chlorides", "freesulfurdioxide", "totalsulfurdioxide", "alcohol", "sulphates", "ph", "density",
      "acidindex", "citricacid", "labelappeal", "volatileacidity", "residualsugar", "stars"] # trial 4 - added stars again
```

Before testing various models, I ran the Negative Binomial, Poisson, and Linear models for different variable options to determine which produced the best AIC/BIC and R-Squared values. The params named p0 were the given sample code params. They are not included in the comparison. Here are the full results for each.

	Neg Binom	**Poisson**	**Linear**
P1 (AIC/BIC, R^2)	51530 / 51545, 0.052	51530 / 51545, 0.060	50003 / 50018, 0.774
P2 (AIC/BIC, R^2)	53564 / 53616, 0.014	53808 / 53860, 0.018	52306 / 52358, 0.729
P3 (AIC/BIC, R^2)	51600 / 51697, 0.051	51598 / 51688, 0.059	50244 / 50333, 0.200
P4 (AIC/BIC, R^2)	50543 / 50648, 0.070	50541 / 50638, 0.078	48963 / 49060, 0.276

Based on these values, it appears that the more complex models involving most of the variables are better performing. The other trend I see is that those models which included **labelappeal** and **stars** also show a better fit than those without them. As I test additional classification models, I will only consider *p3* and *p4* variable sets. I also found it interesting that variables which were not statistically significant with fewer variables (like ph and sulphates) became significant again when other variables were added along with them in more complex models **p3** and **p4**. The variable **fixedacidity** remained statistically insignificant throughout all models and was therefore not included in any of them.

Focusing for a moment on the p4 model I get the following summary output for the Poisson regression.

```
                          Results: Poisson
==========================================================================
Model:                  Poisson           Pseudo R-squared:  0.078
Dependent Variable:     target            AIC:               50541.3532
Date:                   2018-08-19 23:54  BIC:               50638.2918
No. Observations:       12795             Log-Likelihood:    -25258.
Df Model:               12                LL-Null:           -27401.
Df Residuals:           12782             LLR p-value:       0.0000
Converged:              1.0000            Scale:             1.0000
No. Iterations:         9.0000
--------------------------------------------------------------------------
                     Coef.   Std.Err.     z     P>|z|   [0.025   0.975]
--------------------------------------------------------------------------
chlorides           -0.0606   0.0164  -3.6841 0.0002  -0.0928  -0.0284
freesulfurdioxide    0.0143   0.0035   4.0655 0.0000   0.0074   0.0212
totalsulfurdioxide   0.0110   0.0023   4.8465 0.0000   0.0065   0.0154
alcohol              0.0547   0.0141   3.8795 0.0001   0.0271   0.0824
sulphates           -0.0190   0.0057  -3.3191 0.0009  -0.0303  -0.0078
ph                  -0.0249   0.0076  -3.2661 0.0011  -0.0399  -0.0100
density             -0.4470   0.1920  -2.3275 0.0199  -0.8234  -0.0706
acidindex           -0.1247   0.0044 -28.3158 0.0000  -0.1333  -0.1161
citricacid           0.0142   0.0059   2.4170 0.0156   0.0027   0.0258
labelappeal          0.2002   0.0060  33.2984 0.0000   0.1884   0.2119
volatileacidity     70.5610  10.3927   6.7895 0.0000  50.1917  90.9302
residualsugar      -70.5446  10.3942  -6.7869 0.0000 -90.9168 -50.1723
stars                0.2128   0.0065  32.7981 0.0000   0.2001   0.2255
==========================================================================
```

Looking at the coefficient values, we first want to make sure that the p-values reported show all coefficients are significant, which they are as we see none above 0.05 above (note this was not the case for some of these same variables when using different combinations of variables). Also despite a lack of knowledge about the contributions of each of these attributes to overall wine quality we can spot check a few to assess if they make sense on the surface. Labelappeal, stars and alcohol all seem to have a reasonably positive influence on the response. Also common variables such as the two sulferdioxides and the two acidity variables have similar coefficient values.

I then moved on to try several more models using the same set of variable parameters. These additional models consist of classification models and regression models which can be considered for their ability to produce a prediction of the number of rings. They are ranked by MSE/MAE and Accuracy Score below. The sklearn methods for scoring based on each model's predicted values were used however the primary metric initially will be the MSE, more explanation of the tradeoffs are listed below.. The NAs in the table below indicate that they were not scored. I decided to focus only on models defined by variable sets p3 and p4 for the rest of the report.

Model	Neg Binom	Poisson	Linear	Ran Forest	SVC	Grad Boost	Dec Tree	KNN
p1	NA	NA	NA	NA	NA	NA	NA	NA
p2	NA	NA	NA	NA	NA	NA	NA	NA
p3 (MSE/MAE) Accuracy	2.984/1.358 0.247	2.984/1.358 0.247	2.966/1.343 0.252	1.713/0.997 0.354	3.894/1.245 0.410	2.512/1.224 0.280	0.0/0.0 1.0	3.864/1.147 0.524
p4 (MSE/MAE) Accuracy	2.665/1.306 0.246	2.665/1.306 0.246	2.683/1.293 0.256	0.964/0.731 0.462	3.799/1.207 0.443	1.451/0.920 0.365	0.0/0.0 1.0	3.329/1.047 0.540

To decide on a final model to submit, I tested three approaches using the same set of variable parameters, p4. I decided on these independent variables due to the analysis above that showed for my first three models the variance explained by the additional variables improved my model performance as compared to using less predictors. The p4 variables are as follows:
p4 = ["chlorides", "freesulfurdioxide", "totalsulfurdioxide", "alcohol", "sulphates", "ph", "density", "acidindex", "citricacid", "labelappeal", "volatileacidity", "residualsugar", "stars"].

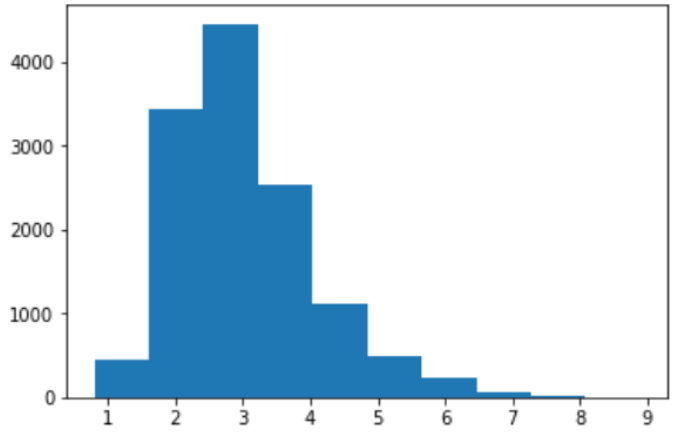
The basis for my scoring of the variable selection was a combination of AIC, BIC and R-Squared scores for each but I will be using a separate metric for the final model selection. The top three models were scored using the Mean Squared Error (MSE), the similar Mean Absolute Error (MAE) and the calculated accuracy score from the model predictions. The MSE and MAE measure the error associated with the difference between the observed and the predicted target values but place different weights on things like outliers or differences in data scaling. I'm looking for the overall lowest values for both MSE and MAE as I compare my potential models. The accuracy score basically measures the amount of correctly predicted observations as a percentage of the total. This is also sensitive to scaling or uncleaned data.

From the public Kaggle scores I noticed there seems to be a balanced relationship between the MSE and the Accuracy score that determines to the overall best fit score. Case in point: while the errors are lower for the Random Forest model, the Gradient Boosting model gave a better score on the predicted values in Kaggle. Overfitting may be the issue; here are the comparisons of the top four models.

Model 1 (Neg Binom & Poisson using model parameters p4):
I decided to start with Poisson given the performance in my initial trials between Negative Binomial and Poisson were nearly identical for a given set of parameters. The Poisson errors and accuracy values were the same as Negative Binomial and the AIC/BIC and R-Squared values for each set of variables were all very close to Negative Binomial values as well as shown in the variable comparison table. I am showing the Poisson predictive results here. The R-Squared value here is 0.078.

The Plot:

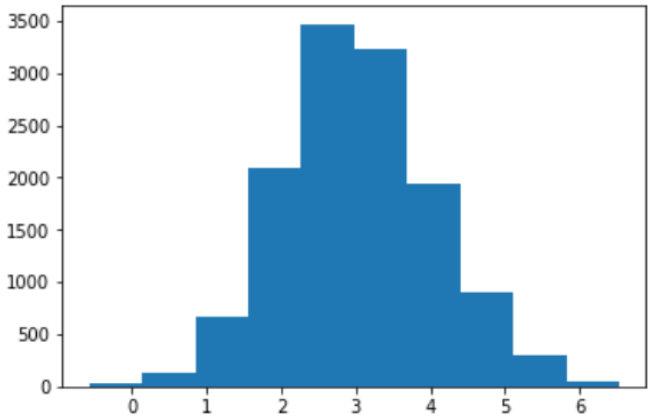
```
The model MSE is 2.665
The model MAE is 1.306
The accuracy score is 0.246
```



Model 2 (Regression using model parameters p4):
I tried linear regression as well. While this model scored slightly better than Poisson in terms of accuracy, the errors were higher. The R-Squared value reported was 0.276. I did not expect the linear regression model to fair this well by comparison however it can definitely be improved on with other regressor models. Based on the overall error I would rank this below Negative Binomial and Poisson.

The Plot:

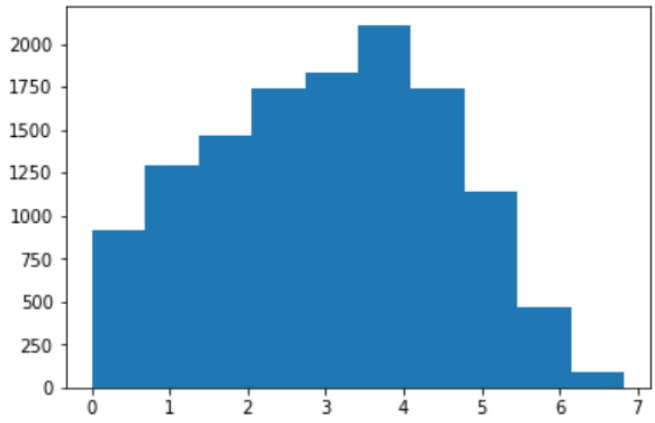
```
The model MSE is 2.683
The model MAE is 1.293
The accuracy score is 0.256
```



Model 3 (Random Forest using model parameters p4):
This model uses the Random Forest Regressor from the ensemble methods of scikit. The default parameters of 10 estimators, using MSE criterion, and other built-in defaults were used. The resulting R-Squared score for this model is 0.7401. This model gave a nice improvement over the first three. With lower overall MSE and MAE errors and a much higher R-Squared value, this seems to be the best so far.

The Plot:

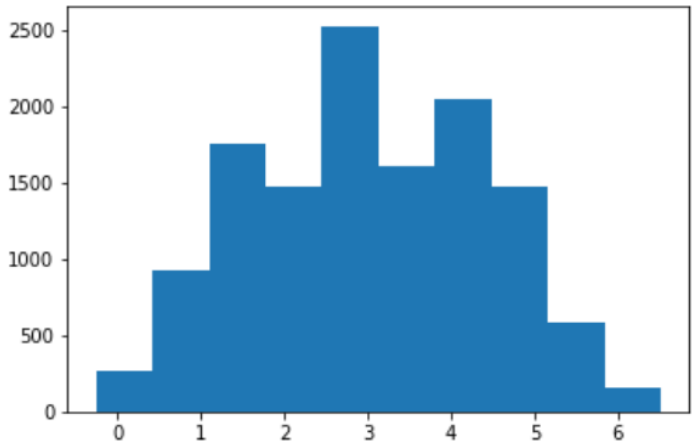
```
The model MSE is 0.964
The model MAE is 0.731
The accuracy score is 0.462
```



Model 4 (Gradient Boosting using model parameters p4):
While the reported MSE and MAE values are slightly higher than the Random Forest model, this is still an improvement over the initial three (Neg Binom, Poisson and Linear Reg) models. The R-Squared value reported was 0.6089 which was slightly lower than the Random Forest model. However, this Gradient Boosting Regression model also gave the best public score when submitted to Kaggle of 1.2799 despite being improperly label as gradient descent.

The Plot:

```
The model MSE is 1.451
The model MAE is 0.92
The accuracy score is 0.365
```



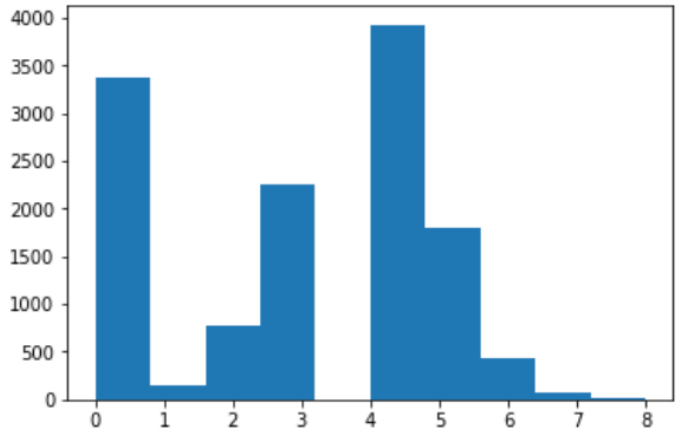
As I compared the results of several of these models it seems the regression models outperform the classification models. While it may be very dependent upon the parameters used, the classification approaches like SVC and KNN did not perform as well as I had initially expected. Before deciding on my

final 'best' model I wanted to investigate the difference between regression and classification in the ability to predict the number of wine cases. This is described in the section that follows and uses information obtained from the scikit learn website that covers the Ensemble Methods.

Select Model

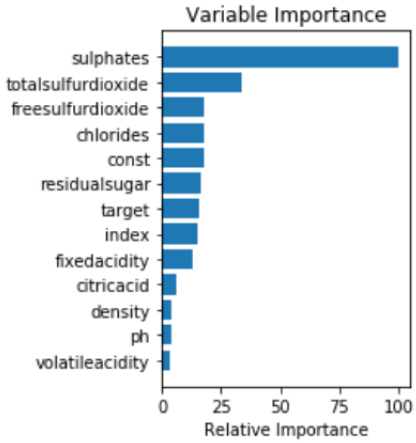
The Gradient Boosting Regression model gave the best public score and has one of the lowest error metrics of model options. However before making a final model selection, I tried the similar Gradient Boosting Classification model. Using similar parameters as the regression model, I fit the p4 variables to the target response to compare the same three metrics to the regression output. The resulting Gradient Boosting Classification output is given here.

```
The class model MSE is 1.954
The class model MAE is 0.751
The class accuracy score is 0.584
```



This is interesting because while the MSE and MAE errors are actually higher than the regression model, the accuracy score is also higher, and therefore better. The resulting distribution is much different than was obtained by the other regression models and is zero-inflated as shown in the plot above. Unfortunately, the predicted file scored worse on the Kaggle public data set which indicates the lower error metric doesn't necessarily mean a better fitting model. Incidentally, the R-squared score given for the Gradient Boosting Classification model is 0.5844 which is worse than the Gradient Boosting Regression model. Thus it seems the R-Squared value must be considered along with the MSE/MAE and Accuracy values to determine the best selection. I also ran a few quick tests of the Random Forest Classifier to compare to the Random Forest Regression and a similar story emerged. The Classifier scored worse than the Regression on all fronts.

In the end, I chose to stick with the Gradient Boosting Regression model defined by the following model string: "**target ~ chlorides + freesulfurdioxide + totalsulfurdioxide + alcohol + sulphates + ph + density + acidindex + citricacid + labelappeal + volatileacidity + residualsugar + stars**"



Just for fun, I also tried running a sorted plot of the relative feature importances using the builtin sklearn module on the fitted final model. The results were a bit surprising, assuming I arranged and plotted the feature variables correctly.

Sulphates and **totalsulfurdioxide** seem to have the most relative importance according to this regression fit while **ph** and **volatileacidity** have the least. Notable omissions here are labelappeal and stars but again I'm not familiar with the algorithms ability to accurately report these.

Scored Data File

The final scored data is named **hw03_andrewknight_p4_gradboost_finalpreds.csv** and contains the same predictions used for the best scoring Kaggle submission file. The file contains the index column and the p_target which lists the predicted wine case counts for the final selected model. These values were left in the float format as un-rounded values greater than zero.

Conclusion

In conclusion, it is easy to see how a multitude of model options can complicate the selection process when faced with regression or machine learning task. As always it is important to understand the data set as best as possible and perform EDA as a due diligence even if the specific subject matter is not clear to us. This was certainly the case here for me as the contributions of each of the components within the wine chemistry/characteristics are totally unfamiliar.

It was also interesting to see the comparison between models and how choosing a single metric doesn't always provide a consistent means for establishing the best model. It seems to require an honest assessment of several factors to determine the right model and often the 'right' model doesn't perform the best on the original training set as was the case when comparing my Random Forest Regression model to my Gradient Boosting Regression model. This assignment also helped me appreciate cases where there is a relatively small number of possible variables to use as predictors.

As a final observation about the outcome of the four final models reviewed, I thought it was interesting how the predicted values produced relatively similar scores (based on the MSE/MAE and accuracy values) but they produced strikingly different prediction count distributions. Comparing the plots in the four models shows that there can be quite a lot of variation in the predicted counts depending on the model even among those that perform similarly. As such, it would be wise to take this into consideration when deploying any of these in production and making business decisions off of them. Now, I'm going to go enjoy a simple glass a wine. Cheers!