Predicting Wine Quality using Random Forest

Introduction

In this project, I attempt to create a random forest model that can accurately predict the quality of a given wine, when the quality is rated from 1-10. The dataset gives 12 variables; One is the quality of a wine, a rating from 1-10, that I chose as my response. The other 11 variables are mostly chemical properties, such as alcohol content, citric acid content, acidity, and pH, among others. Using this data, we use random forest to learn how to "classify" each wine into one of 7 categories; in this case, the values 3-9 for their quality (no wines were rated with a quality of 1,2,or 10). We create a model that is accurate approximately 70% of the time in its predictions of quality. 70% accuracy means that our model could reasonably be used to recommend wines to a consumer or determine what properties of a wine lead to high quality, as being wrong in this case would not lead to any dire negative consequences.

Dataset

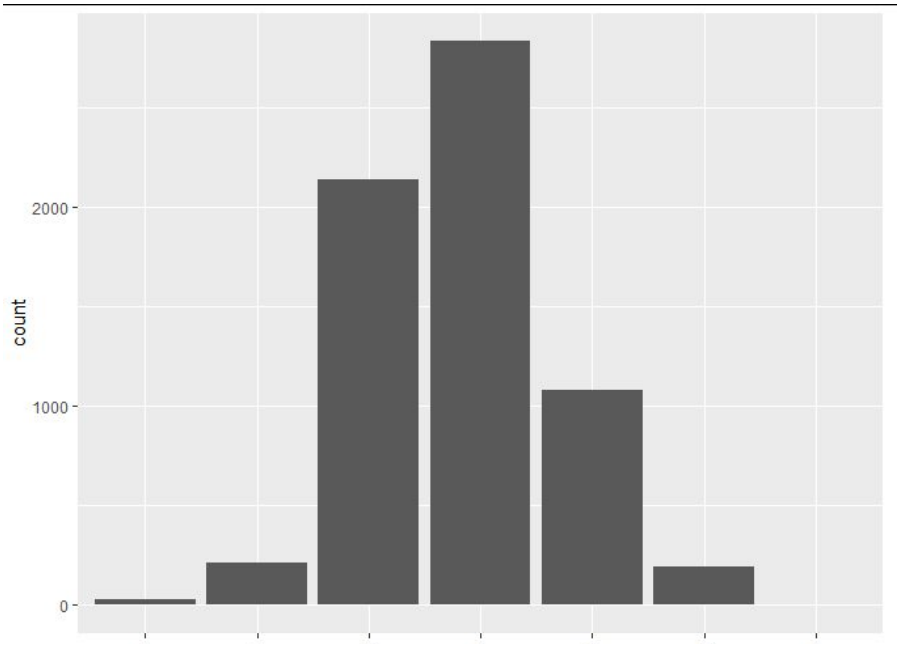
I chose to use the wine quality dataset from UCI: https://archive.ics.uci.edu/ml/datasets/Wine+Quality for this project. The dataset consists of 6947 different observations of wines, split into two sets: red wine(with 1599 observations) and white wine(with 4898 observations), which I combined into a single dataset. There are 12 attributes, one of which is quality, our response variable. This is a rating from 1-10. The other 11 are continuous numeric variables that are: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol content. The dataset is cited below.

Reference

P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis.
Modeling wine preferences by data mining from physicochemical properties. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.

Model Selection and Validation

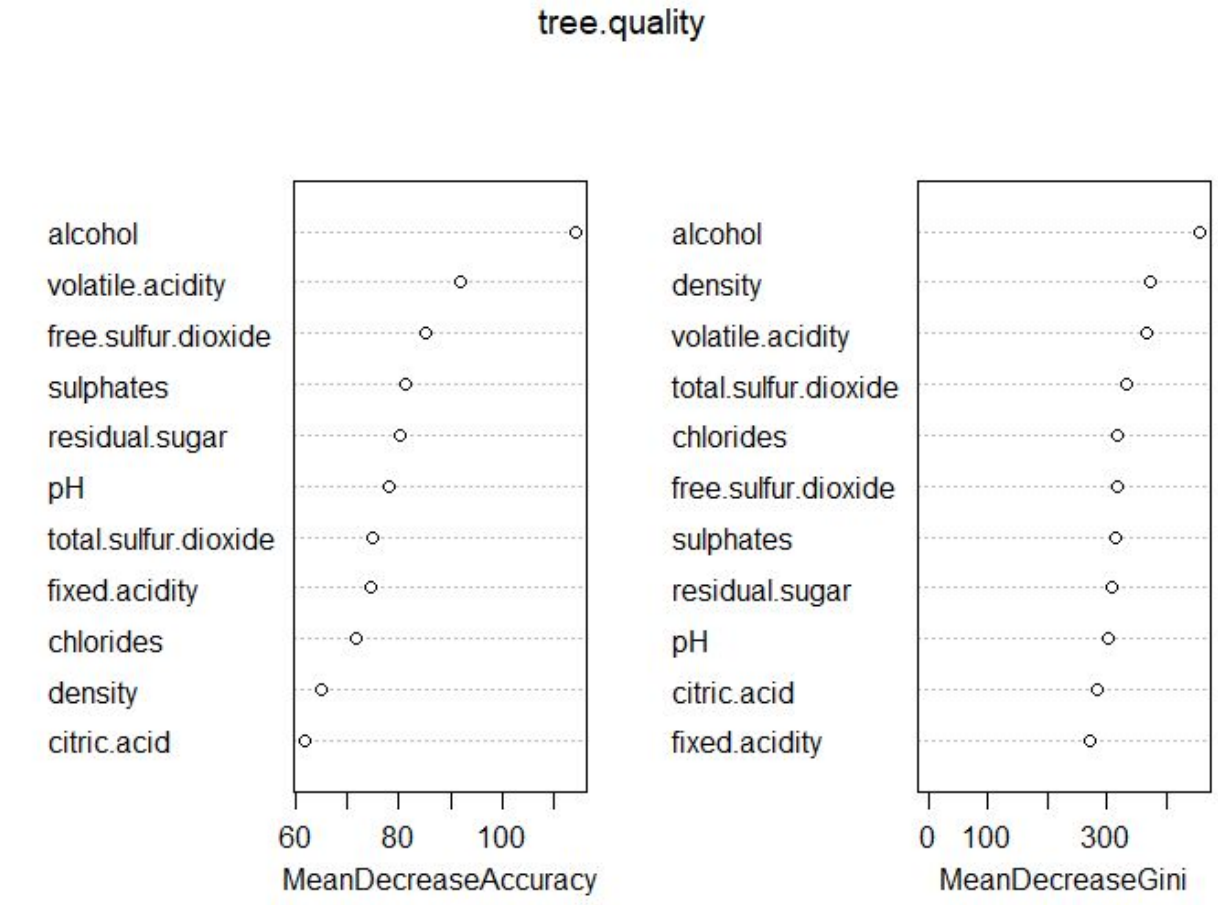
Upon examining the problem, I decided that it was, in essence, a classification problem; which of these 7 categories (quality ratings) did a given wine fit into? I decided then to use a random forest model to build a classifier. Before getting started, I created a bar chart to see how many wines fit into each category of quality:



Given that the majority of wines were in the 5-7 category, and so few fit outside of that range, I opted to use the validation set approach. On top of its simplicity and ease of implementation, I didn't think that training models on the even smaller subsets of data that would be created in k-fold and LOOCV would be a good idea, given that those models would be more easily skewed by the wines outside the middle range. I create a test set containing 1/6th of the approximately 6500 observations and a training set containing the rest. We then train a random forest model with 500 trees and mtry = sqrt(P) to solve the classification problem. We find that the OOB error estimate is 30%, meaning our model fits the data reasonably well, but not fantastically.

Model Interpretation

Using a variable importance plot, we can see which attributes are most important in determining a wines quality:



From this, we see that alcohol is the most important factor for determining a wine's quality, followed by its volatile acidity.

We can begin to assess the overall accuracy of our classifier using the models OOB error estimate and confusion matrix:

```
Call:
 randomForest(formula = quality ~ ., data = allWine, mtry = sqrt(P),      importance = TRUE, subset = train)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 30.01%
Confusion matrix:
  3  4    5    6   7  8 9 class.error
3 0  1   15   11   1  0 0   1.0000000
4 1 29   91   58   3  0 0   0.8406593
5 0  5 1325  465  14  0 0   0.2675511
6 0  4  339 1864 122  2 0   0.2003432
7 0  0   21  368 508  2 0   0.4349277
8 0  0    1   50  48 64 0   0.6073620
9 0  0    0    1   2  0 0   1.0000000
```

To find that we have an OOB error rate of 30.01%. We compare these results to our test data predictions and the "balanced accuracy" field from the caret package's confusion matrix function:

```
yhat.tree   3   4    5    6   7   8   9
        3   0   0    2    0   0   0   0
        4   0   2    1    0   0   0   0
        5   2  20  250   78   6   1   0
        6   0  12   74  388  80  11   2
        7   0   0    2   39  93   8   0
        8   0   0    0    0   1  10   0
        9   0   0    0    0   0   0   0
> |
```

```
Statistics by Class:

                     Class: 3 Class: 4 Class: 5 Class: 6 Class: 7 Class: 8 Class: 9
Sensitivity          0.000000 0.058824   0.7599   0.7683  0.51667 0.333333 0.000000
Specificity          0.998148 0.999046   0.8579   0.6898  0.94568 0.999049 1.000000
Pos Pred Value       0.000000 0.666667   0.7003   0.6843  0.65493 0.909091      NaN
Neg Pred Value       0.998148 0.970343   0.8910   0.7728  0.90745 0.981326 0.998152
Prevalence           0.001848 0.031423   0.3041   0.4667  0.16636 0.027726 0.001848
Detection Rate       0.000000 0.001848   0.2311   0.3586  0.08595 0.009242 0.000000
Detection Prevalence 0.001848 0.002773   0.3299   0.5240  0.13124 0.010166 0.000000
Balanced Accuracy    0.499074 0.528935   0.8089   0.7290  0.73117 0.666191 0.500000
```

And we can see that we achieve an overall average balanced accuracy of approximately 64% on our training data; 6% less than our OOB estimates.

Conclusion

We manage to predict a wine's quality using random forest with roughly 70% accuracy. Examining our model, we can see that alcohol and acidity are the most important factors in determining a wines quality from this dataset. The classifier presented here is far from perfect, and could likely be tweaked to achieve higher accuracy.