Brian Miller
06/01/2022
COMP 4441 Final Project

Using a Random Forest Classifier To Predict Wine Tasting Scores

Introduction

For my final project I wanted to build a model that could correctly predict wine quality based on various chemical signals and measurements. Before taking COMP4441 I had heard of Random Forests (RF) and their application for classification and regression problems, but I hadn't had the chance to implement one myself. Thus I took this opportunity to learn the basics of how RF models work, and how to build and evaluate them.

Random Forests Explained

Random Forests is a method for classification and regression problems when there are multiple explanatory variables. RF models use many varied decision trees to capture the direct and indirect correlations between the explanatory variables and the outcomes. When a single decision tree is made, first a bootstrapped version of the dataset is used, and only a random sample of your explanatory variables are used when making each branching point. These two factors create the tree to tree variation that allows RF models to be flexible and accurate even with multiple explanatory variables (Figure 1). In order to classify a given new wine using the eleven explanatory variables the new datapoint is run through all of the trees in the forest, the most common output is the RF's output (Figure 2).

Data Source and Main Features

The dataset used for this project came from a paper out of the University of Minho, Portugal that sought to model wine preferences from physicochemical properties. After testing neural network, support vector machine (SVM), and multiple regression models the authors found that a SVM model that allowed for +/- 1 red wine score error worked the best with an accuracy of 89% (Cortez et al., 2009). With no mention of the authors attempting to use RF models, this seemed like an opportunity to try a different approach.

The data was downloaded from UC Irvine's curated repository of datasets for machine learning. The data consisted of a single table with 1600 rows, each containing data on a specific portuguese red wine variant. Along with the wine's quality score (the median score from three professional wine tasters) the table had eleven other columns with measured physicochemical properties (Figure 3). The red wine scores were heavily centered around scores of 5 and 6, with some exceptions on either side (figure 4).

Parameter Tuning and Variable Selection

There are three main parameters to tune when building a Random Forest. First, is the number of explanatory variables to use at each branch point when building trees. Second, is the number of trees to build. Third is node size, which is the minimum number of data points at each terminal node; this parameter controls tree depth. The first parameter was automatically tuned using functions from the randomForest package (set to 2). The number of trees and the node size parameters were manually determined after testing many different values and picking the ones that minimize the out of bag error of the model. This error metric is calculated by running the data that was excluded during bootstrapping through the model, and seeing how well it predicted the known outcomes, or wine scores. A node size of 2 and 500 trees were the parameters that minimized this error metric (Figure 5).

In addition to the standard RF parameters to tune I was curious if specific explanatory variables were more or less important at predicting wine quality. I ran a Anova test and citric.acid and residual.sugar were insignificant (Figure 6). I later reran the RF model excluding these two variables but the model's precision and accuracy scores were worse. My best guess is that these two variables are indirectly correlated with wine score, and while not captured by the Anova, the RF model uses those indirect relationships to help make more accurate predictions. Thus, all eleven variables were included in the RF models.

Data Requirements

There are three main data requirements for Random Forests models. First, there should be no missing data/NAs. Second, there should be relatively clean data with no obvious outliers. Third, the outcomes should be balanced. Thankfully since the dataset was curated by UC Irvine's ML department the first two requirements were already satisfied. While no analysis was done to identify potential outliers, anecdotally the frequency distributions of the eleven predictor variables seemed clean enough for me to continue with the RF model (Figure 7).

The third requirement for RF models is that the outcomes need to be balanced. After trying multiple different binning approaches the simplistic binary outcome model (good wine or bad wine) had the best load balancing, and the best results (Figure 8). In order to test the precision of the different models I split the data into training and testing subsets, 80% and 20%, respectively.

Results

Using the binary classification RF model I was able to achieve a precision score of 81.34% and 87.06% for the bad wine and good wine prediction, respectively. The five most important variables of making an accurate RF model were: alcohol, sulfates, volatile.acidity, total.sulfur.dioxide, and density (Figure 9). Interestingly, alcohol content was positively correlated with wine score, while volatile.acidity was negatively correlated with wine score (Figures 10 and 11). Volatile.acidity refers to the amount of acetic, lactic, and formic acid in the wine, which usually have a pungent smell, such as vinegar or nail polish removal.

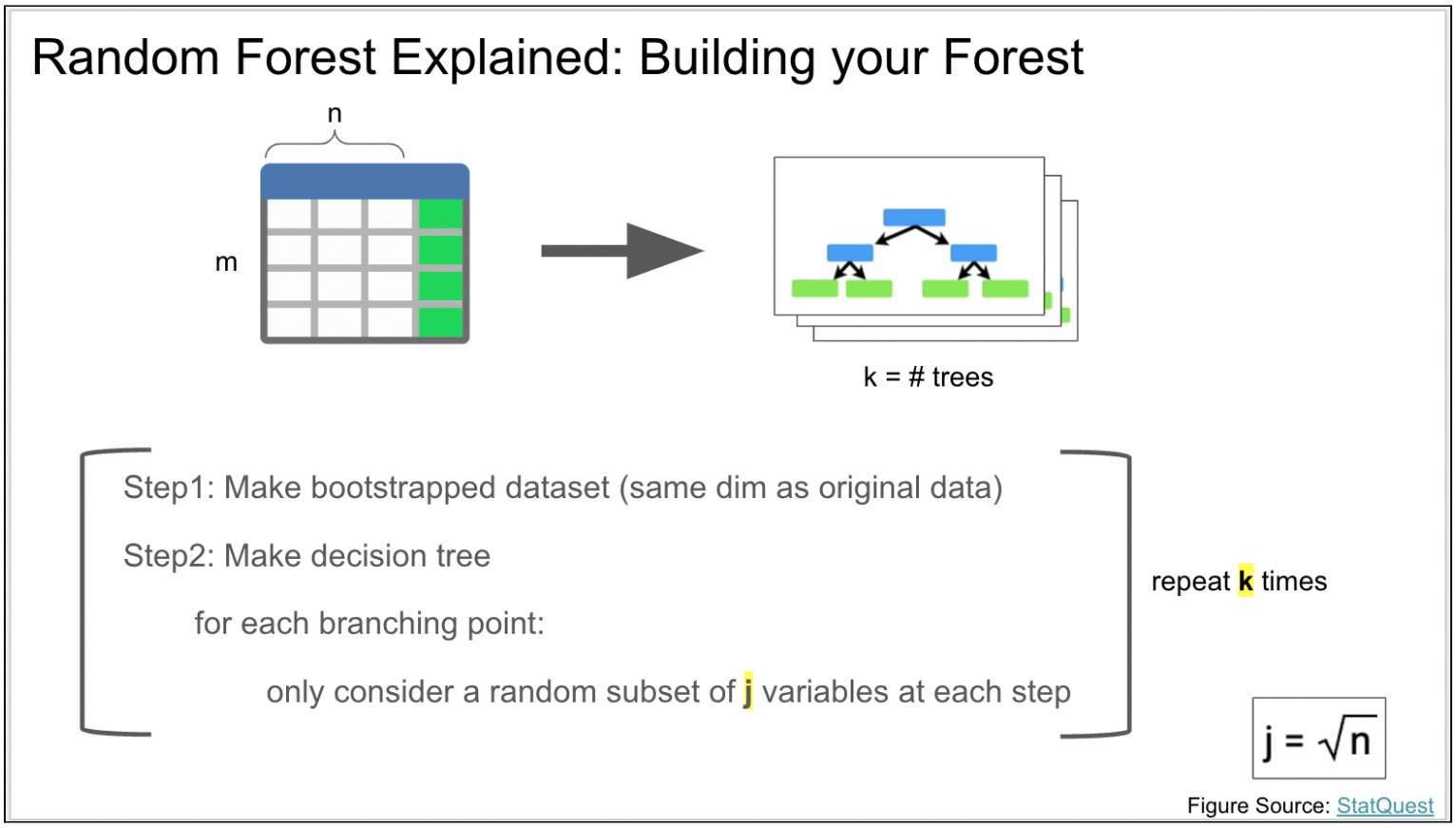


Figure 1: A quick explanation of how Random Forests are built.

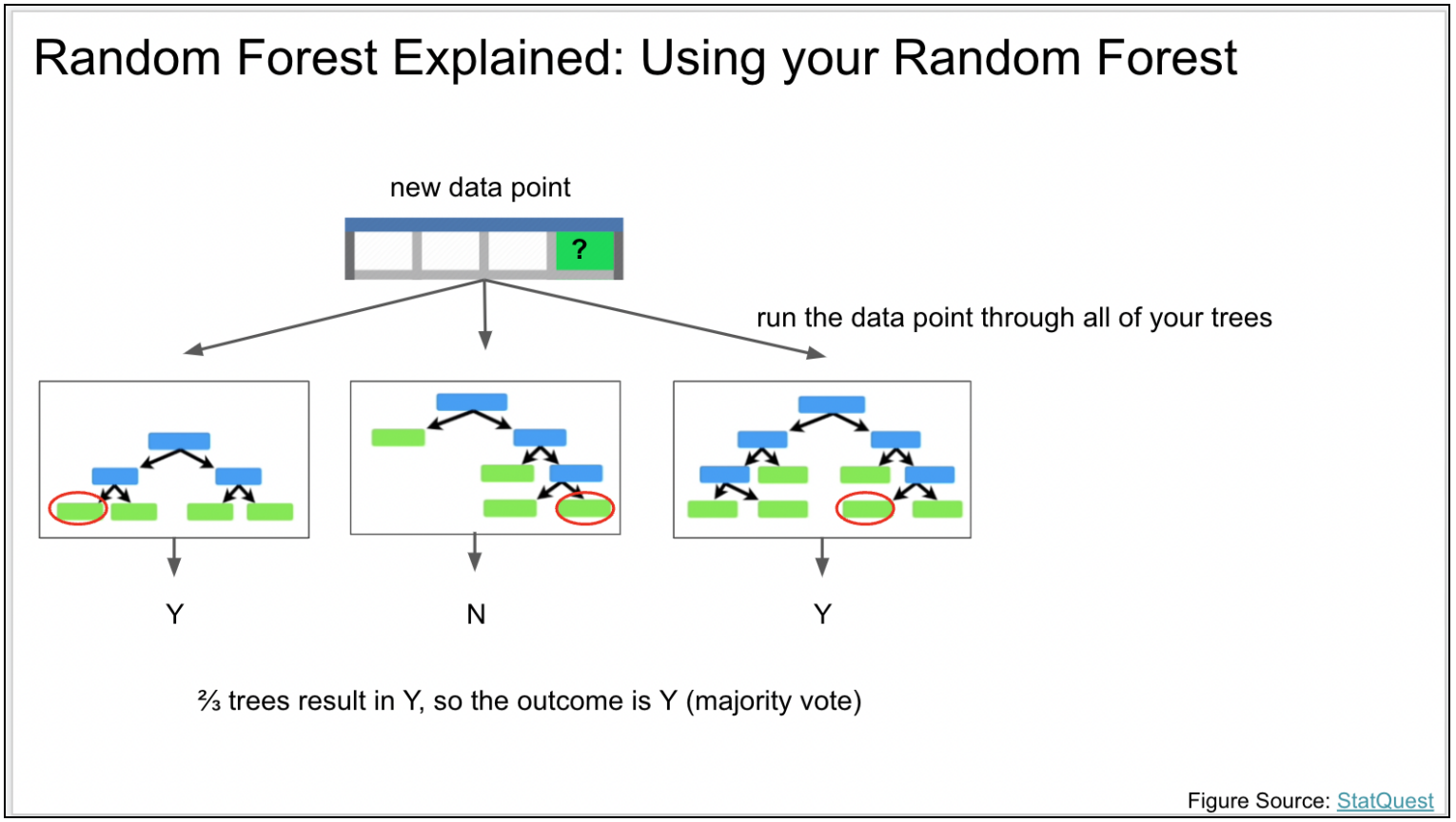


Figure 2: A quick example of how Random Forests are used.

	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality
	34	0.9978	3.51	0.56	9.4	5
	67	0.9968	3.20	0.68	9.8	5
	54	0.9970	3.26	0.65	9.8	5
	60	0.9980	3.16	0.58	9.8	6
...	34	0.9978	3.51	0.56	9.4	5
	40	0.9978	3.51	0.56	9.4	5
	59	0.9964	3.30	0.46	9.4	5
	21	0.9946	3.39	0.47	10.0	7
	18	0.9968	3.36	0.57	9.5	7
	...					

Figure 3: Display of a section of the datatable. Full data table is 1600x12.

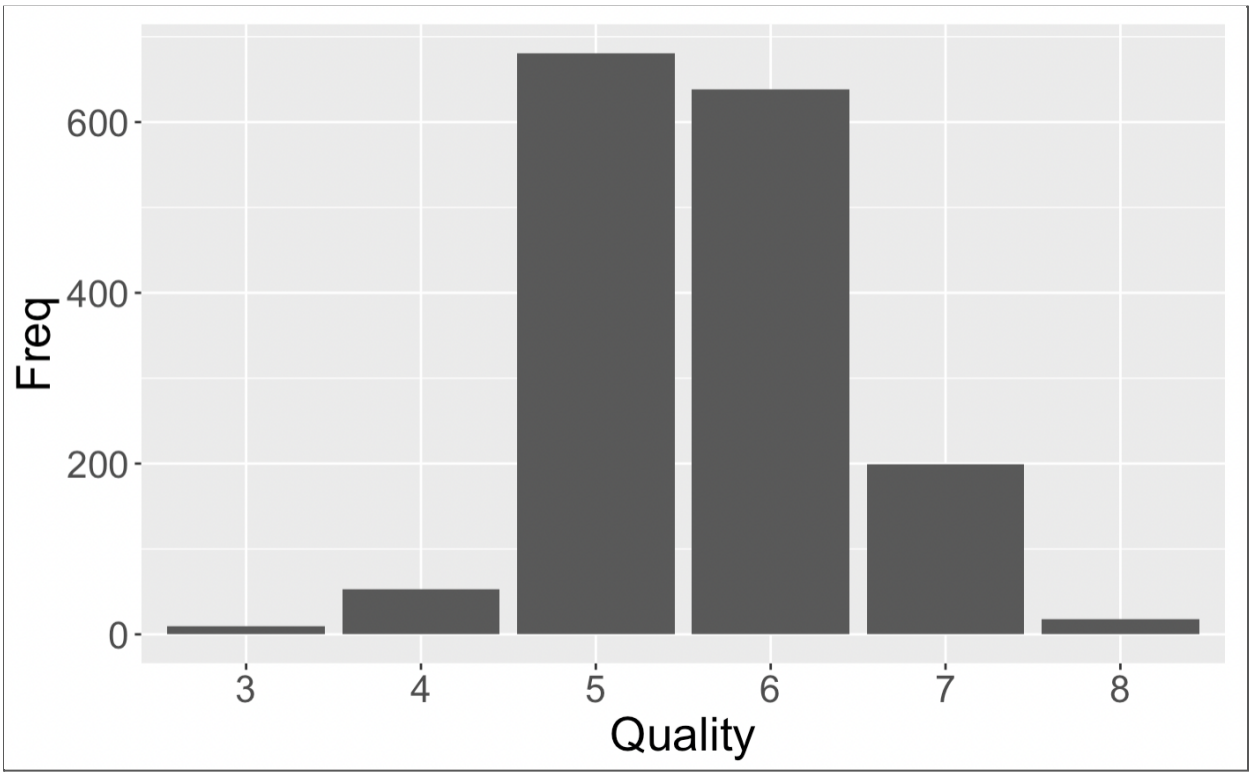


Figure 4: Frequency distribution of red wine quality scores.

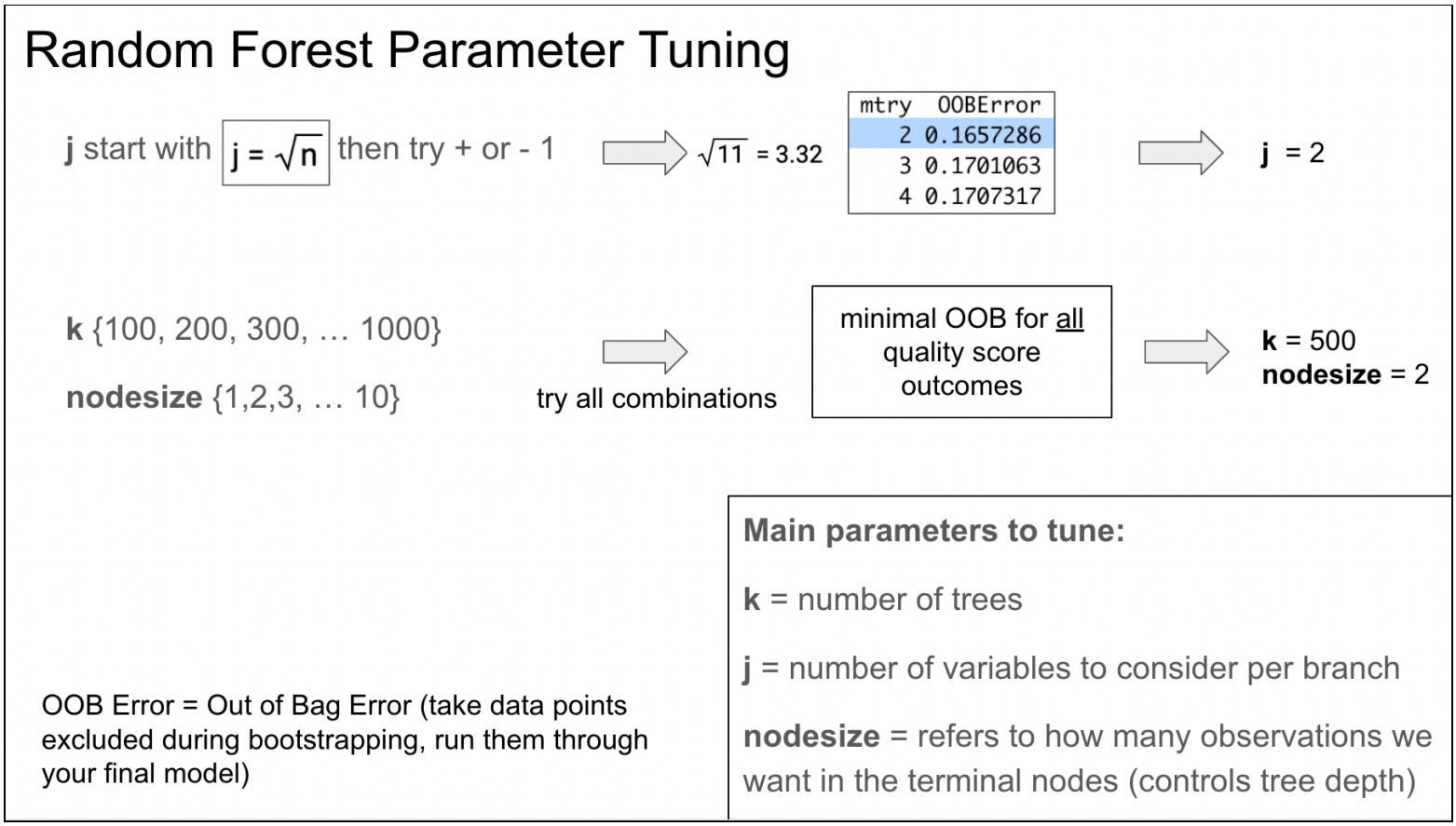


Figure 5: An overview of the Random Forest parameter tuning process

```
                      Df Sum Sq Mean Sq F value   Pr(>F)
fixed.acidity          1   3.60    3.60  20.398 6.75e-06 ***
volatile.acidity       1  37.58   37.58 213.065  < 2e-16 ***
citric.acid            1   0.67    0.67   3.784  0.05191 .
residual.sugar         1   0.00    0.00   0.010  0.92117
chlorides              1   2.80    2.80  15.886 7.03e-05 ***
free.sulfur.dioxide    1   1.58    1.58   8.968  0.00279 **
total.sulfur.dioxide   1  19.70   19.70 111.719  < 2e-16 ***
density                1  16.10   16.10  91.303  < 2e-16 ***
pH                     1   4.51    4.51  25.554 4.80e-07 ***
sulphates              1  17.35   17.35  98.393  < 2e-16 ***
alcohol                1  14.04   14.04  79.637  < 2e-16 ***
Residuals           1587 279.89    0.18
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 6: Anova results. Citric acid and residual.sugar were insignificant when predicting wine score using all eleven predictor variables in a linear model.

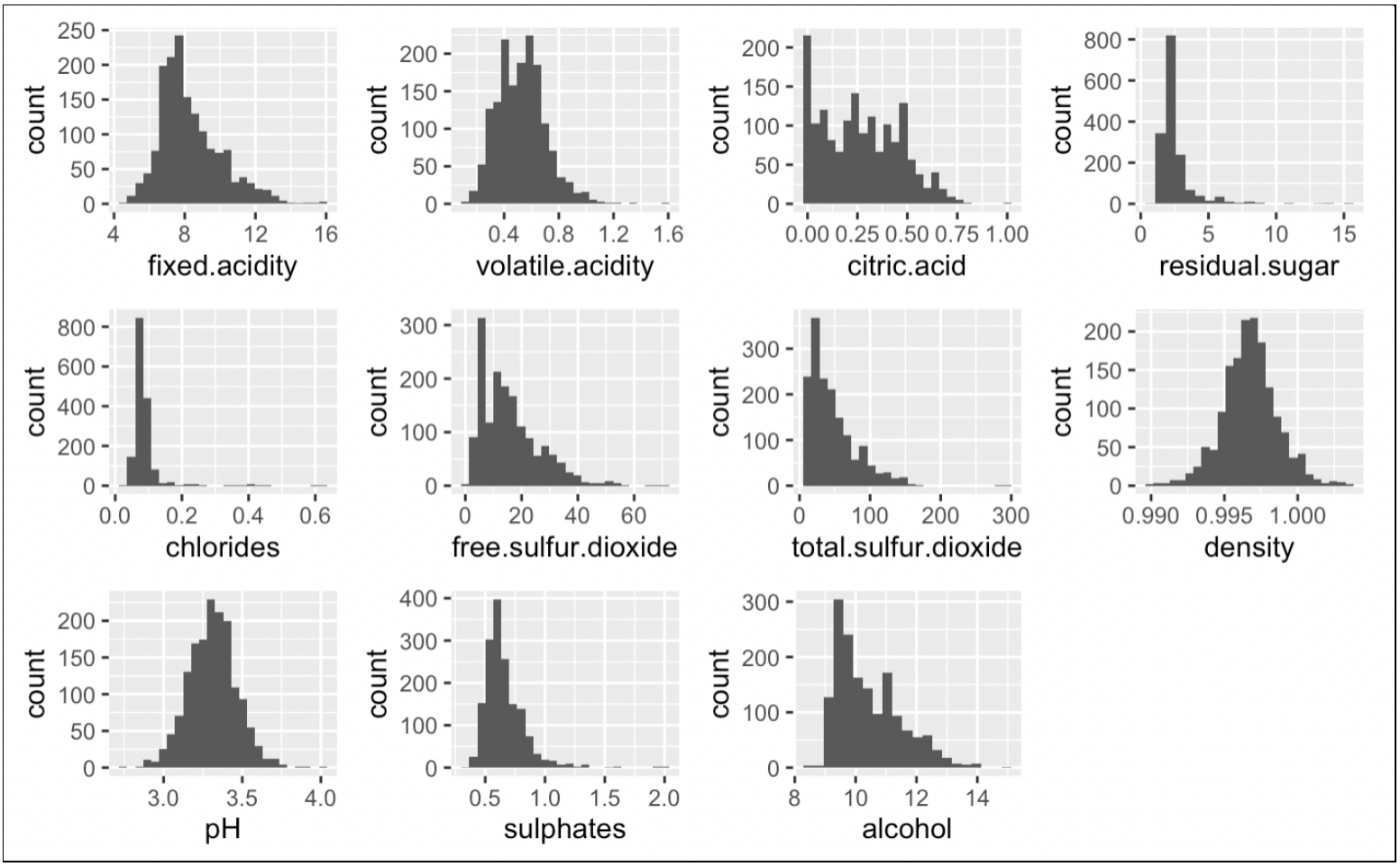
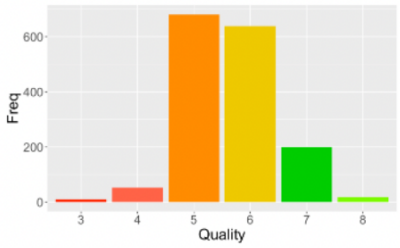
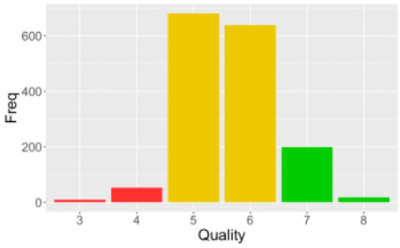


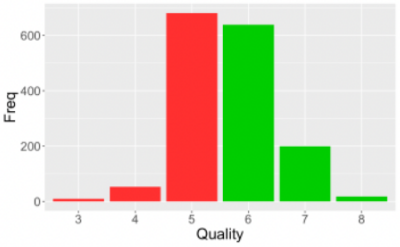
Figure 7: Frequency distributions of all eleven predictor variables



wine_score	n_total	n_training	n_testing	was_correct	was_incorrect	precision
3	10	8	2	0	2	0.0000000
4	53	48	5	0	5	0.0000000
5	681	538	143	114	29	0.7972028
6	638	520	118	90	28	0.7627119
7	199	149	50	28	22	0.5600000
8	18	16	2	0	2	0.0000000



wine_score	n_total	n_training	n_testing	was_correct	was_incorrect	precision
bad(3,4)	63	49	14	0	14	0.0000000
ok(5,6)	1319	1060	259	256	3	0.9884170
good(7,8)	217	170	47	20	27	0.4255319



wine_score	n_total	n_training	n_testing	was_correct	was_incorrect	precision
bad(3-5)	744	594	150	122	28	0.8133333
good(6-8)	855	685	170	148	22	0.8705882

Figure 8: A binary (good/bad) classification approach on how to bin the wine scores resulted in the most accurate Random Forest model.

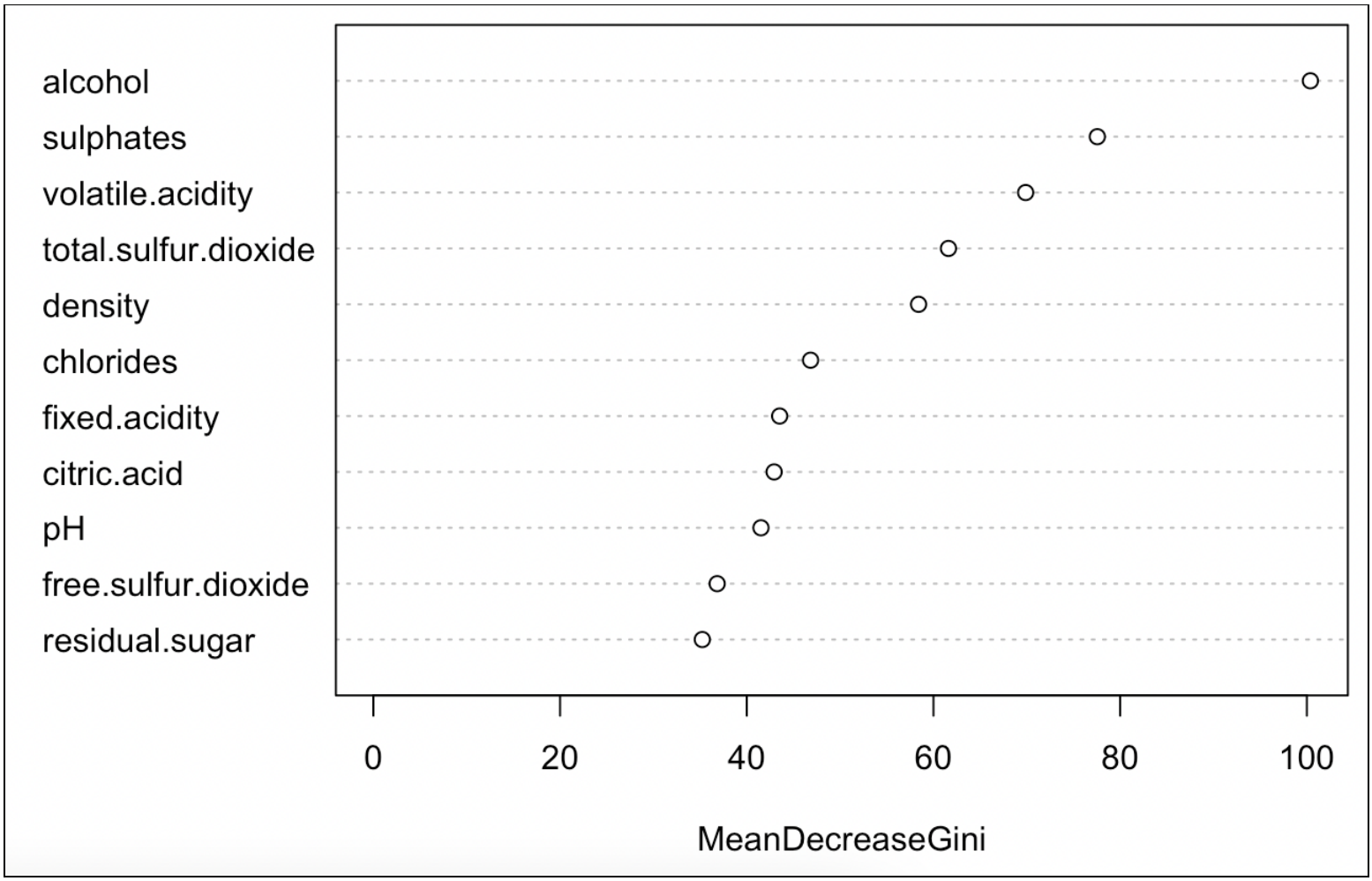


Figure 9: Adding alcohol content to the Random Forest model improved the model performance the most.

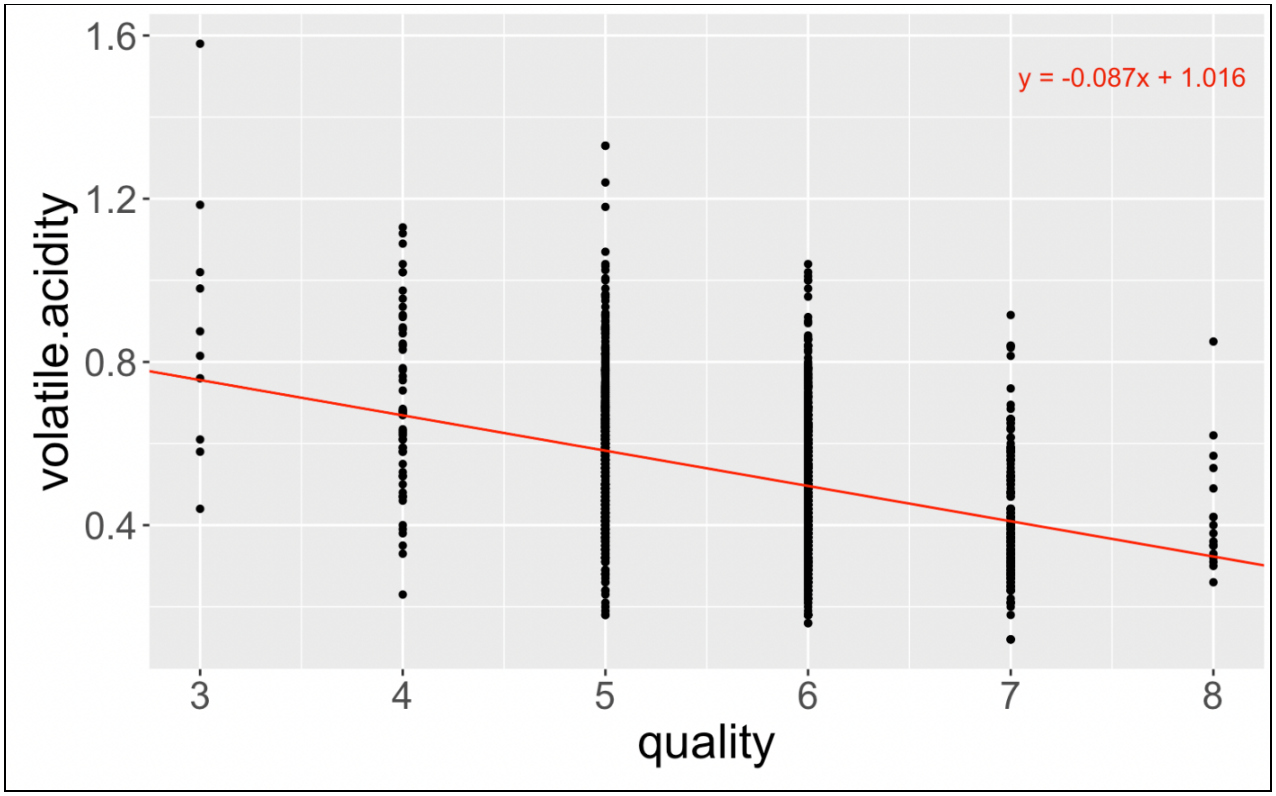


Figure 10: Volatile.acidity is negatively correlated with wine quality. Volatile.acidity was the third most important variable for creating an accurate RF model.

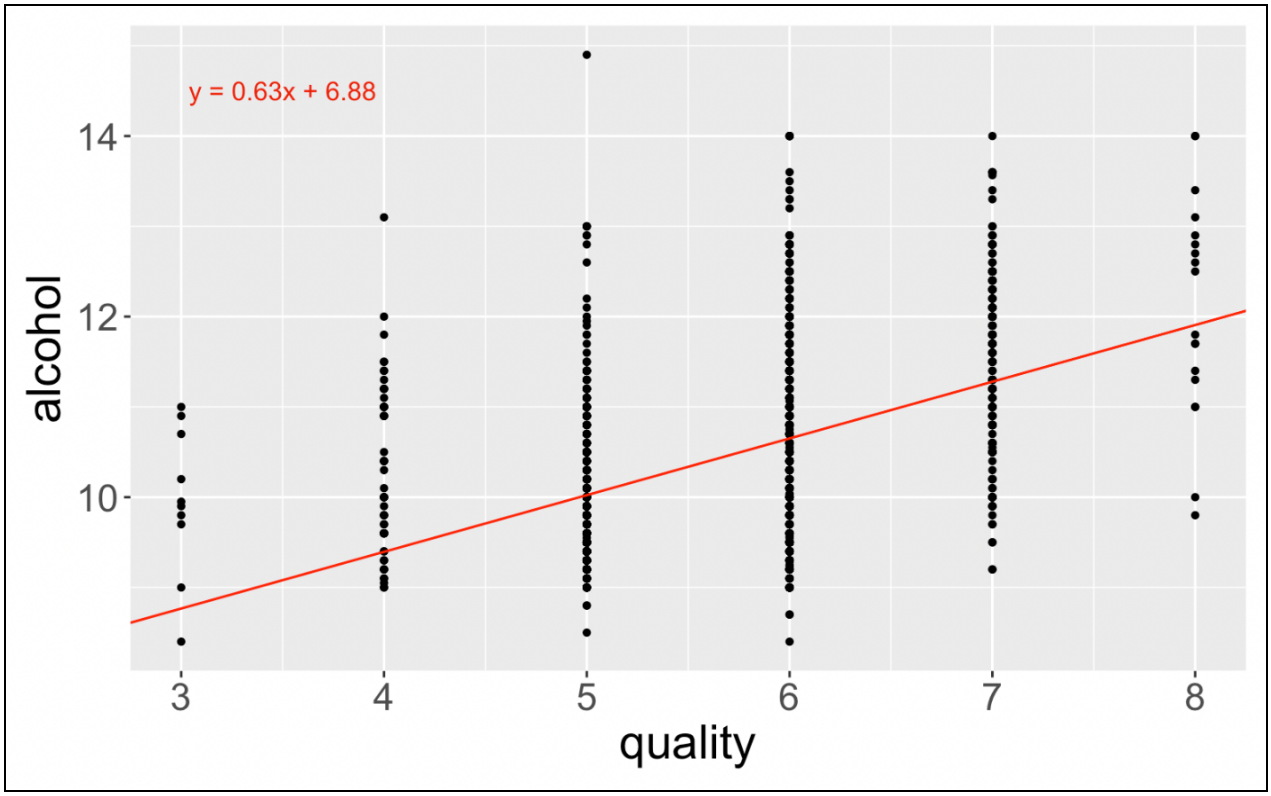


Figure 11: Alcohol is negatively correlated with wine quality. Alcohol was the most important variable for creating an accurate RF model.

References:

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. Pre-Press Release

Most of the visuals used during my presentation to explain how RF models work were taken directly from this youtube video made by StatQuest. Created by Josh Starmer, StatQuest has been a great resource for me and my learning process in the field of data science. Here's a useful link for all of StatQuest's video series: https://statquest.org/video-index/

Link to data (UC Irvine ML repo): https://archive.ics.uci.edu/ml/datasets/wine+quality