# ST563 Final Project

Final Project Group 6

19 April, 2021

## Contents

# Introduction

There has been a push towards the implementation of different analytics solutions within the agriculture industry, especially in the realms of crop health and yield. These sort of analyses can help farmers reduce waste and improve profits. In particular, the viticulture industry stands to benefit from such methods, being that the production of wine is lengthy and multifaceted. Winemakers can work to optimize revenue by analyzing their input and output (crops and wines). Specifically, costs may be cut out if machine learning methods were able to predict the quality of wine based on different metrics, since wine producers would no longer have to hire expert wine tasters to determine wine quality.

In this study, our objective is to try and predict the quality of wine using a variety of statistical learning techniques. We will look at this problem from a regression standpoint as well as a classification standpoint. Using regression, we will try and predict the exact rating (0 to 10) of a particular wine. On the other hand, when we look at classification, we will try to classify a particular wine as either good or bad, taking wines rated from 0-5 as low quality wines and wines rated from 6-10 as high quality wines. We will be assessing the prediction accuracy of the following methods:

Table 1: ML Methods

| **Classification** | **Regression** |
|---|---|
| Classification Tree | Multiple Linear Regression (MLR) |
| Random Forest | Random Forest |

# Required Libraries

To run the code for the project, the following libraries are required:

- `caret`: to do the heavy lifting of training and tuning the models
- `tidyverse`: for all the data reading and wrangling
- `knitr`: for rmarkdown table outputs
- `rmarkdown`: for output documents
- `corrplot`: for correlation structure visualization
- `leaps`: for subset selection
- `rpart`: fitting the basic classification tree
- `rpart.plot`: visualization of the classification tree
- `randomForest`: fitting random forest models

# Data

The data set used is the `Wine Quality` dataset from the UCI Machine Learning Repository. The data set is composed of 1600 observations of different variants of the Portuguese "Vinho

Verde" red wine. For each wine, various physical and chemical features of the wine were measured, including a measurement of quality given by an expert wine taster. Our objective is to use the different features to predict the quality of the wine using different machine learning techniques.

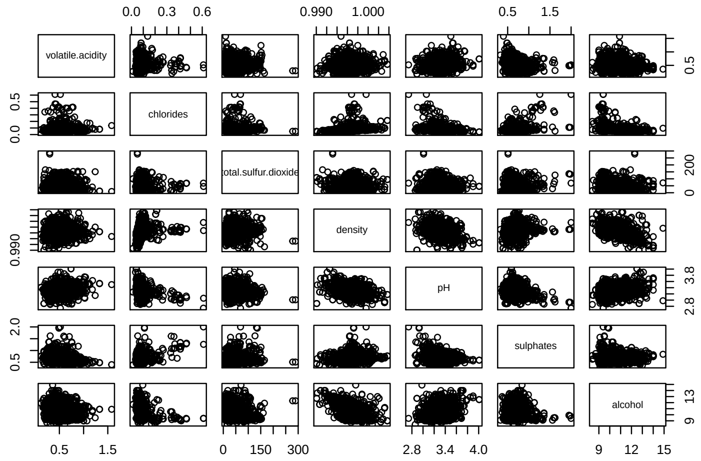
## Variable Descriptions

Each entry in the dataset represents the different metrics of the following attributes of a single type of wine.

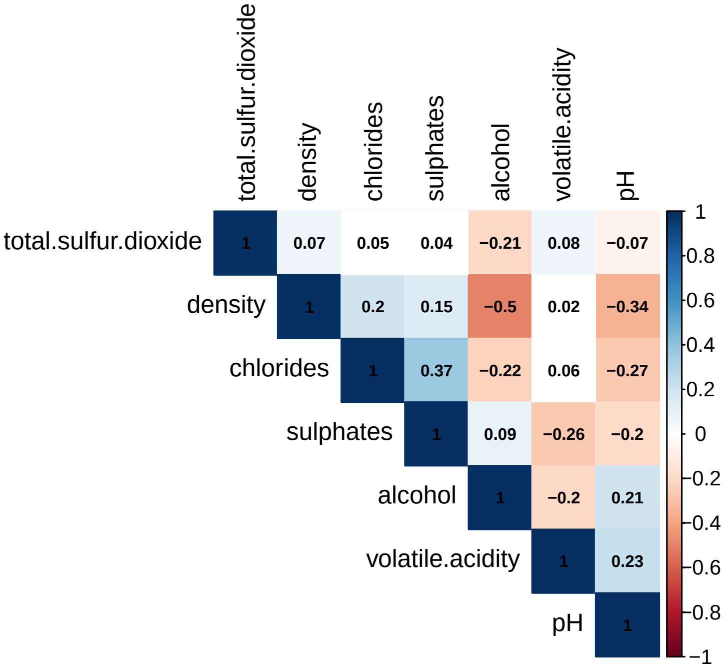
- **fixed acidity**: the quantity of fixed acids found in the wines. The predominant fixed acids found in wines are tartaric, malic, citric, and succinic, all of which come from the grapes with the exception of the succinic acid, which comes from the yeast in fermentation.
- **volatile acidity**: the steam distillable acids present in wine, primarily acetic acid but also lactic, formic, butyric, and propionic acids. These acids generally come from the fermentation process.
- **citric acid**: added to the wine as a natural preservative or for acidity and tartness.
- **residual sugar**: the quantity of sugar left in the wine after the fermentation process.
- **chlorides**: the amount of salt in a wine.
- **free sulfur dioxide**: the amount of $SO_2$ that is not bound to other molecules.
- **total sulfur dioxide**: total amount of $SO_2$ in the wine. Sulfur Dioxide is used throughout all stages of the winemaking process to prevent oxidation and bacteria growth.
- **density**: density of the wine.
- **pH**: pH of the wine.
- **sulphates**: quantity of sulphates in the wine. Sulphates are used as a preservative.
- **alcohol**: alcohol content by volume.
- **quality**: score of quality of the wine given by expert tasters (score between 0 and 10).

## Data Exploration

The first thing that we did with the data upon reading it in was take the numeric variables and create pairwise scatterplots. This way, we can get a feel for what variables are related as well as what the correlation structure looks like, since variables that are very related may cause problems moving forward.



Based on the pairwise scatterplots, we suspect that pH may be linearly correlated with density, sulphates, and alcohol. This leads us to believe that there is some relationship between the chemical properties of the wines. The following correlation plot gives a numeric summary of the linear relationship between the variables.



The only linear correlations that are on the higher side are between alcohol and density, which makes sense because the density of alcohol is slightly less than water. So as the concentration of alcohol gets higher, the density will naturally decrease. Similarly, the sulphates and chlorides seem to have a slightly positive linear relationship, while pH and density have a slightly negative linear relationship.

## Data Processing

Fortunately, the `wine` dataset that we worked with was very tidy; there were no missing values. We didn't standardize the data initially, but when training the support vector machine and the k-nearest neighbors models we centered and scaled the data. We also created the variable for classification, transforming the quality variable into a binary "low" or "high" response as mentioned in the introduction.

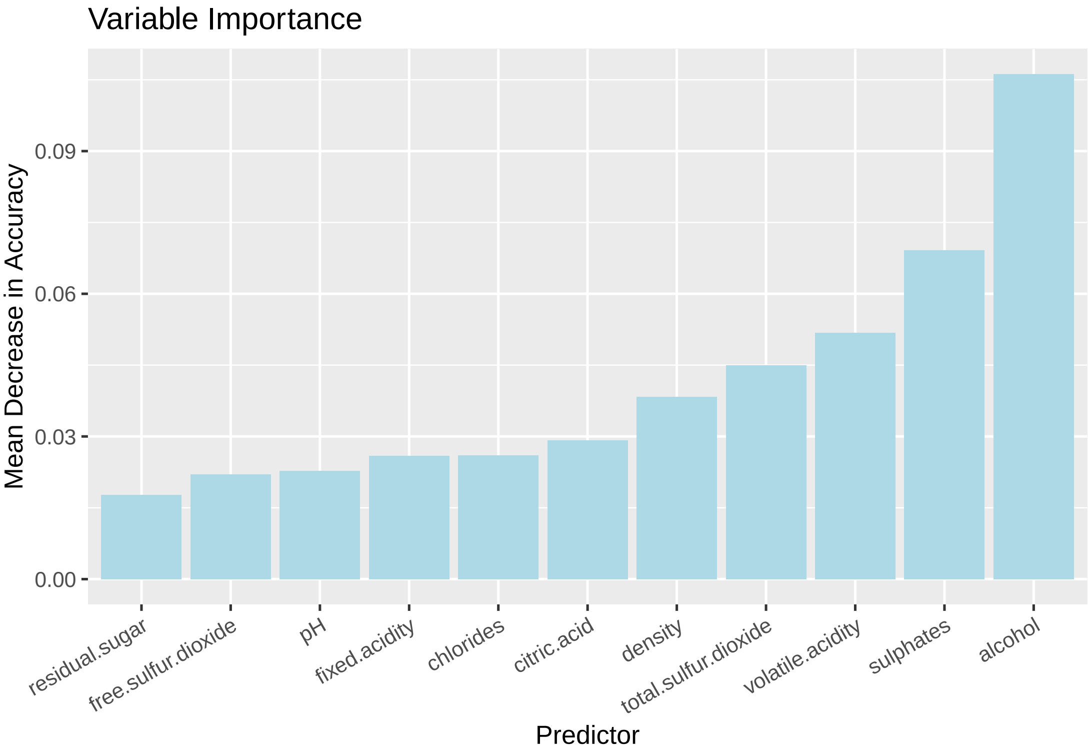
# Methods

## Cross Validation

Again, our focus is on the classification accuracy of machine learning methods. In particular, we will be using traditional cross-validation methods to assess the accuracy, sensitivity, and specificity of each of the models. We split the data into a training and test data set in order to later evaluate the model's prediction accuracy or root mean squared error (RMSE). For this project we used a 75/25 split, training the data on the 75% and testing the trained models on the withheld 25%. We will then repeat this process over 5 folds of the data, averaging the results.

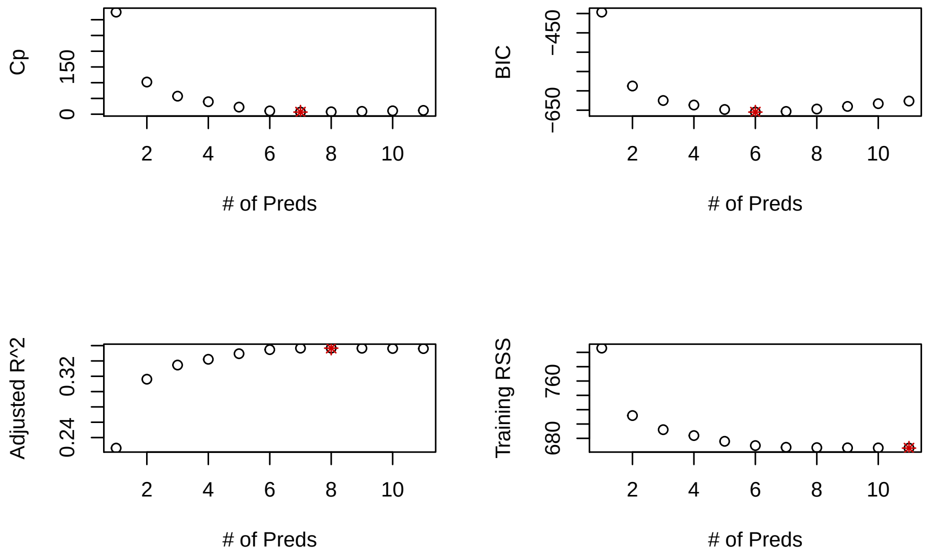
## Variable Selection

Since we have two different tasks that we are trying to complete (classification and regression), we went about selecting the variables for each task differently. For the classification task, we ran an initial random forest model on the entire dataset with selected predictors. We then looked at the variable importance of the predictors to decide which variables we would use in our final models. Within the context of machine learning, variable importance refers to how much a given model "uses" that variable to make accurate predictions. In other words, the more a model relies on a variable to make predictions, the more important it is for the model.



We decided to choose the five variables with the most variable importance, leaving us with `alcohol`, `sulphates`, `volatile.acidity`, `density`, and `total.sulfur.dioxide`. The fact that density proved to be an important variable for classification was relatively surprising. One would think that the wine quality would be determined based on its attributes that are detectable by the human senses. For this same reason, alcohol, acidity, and sulphates made sense in terms of variable importance. Sulphates are essential for the preservation of the wine, an overly- or underly-acidic wine may be considered poor, and the alcohol content has a connection to the fermentation process, where the must (aka the grape juice before it undergoes fermentation) actually becomes wine.

When looking at the problem in terms of a regression context, we used best subsets selection for feature selection. Using the `leaps` library and the `regsubsets()` function in R, we are able to compare all of the possible models from the given set of predictors. Through best subset selection, we can get the set of predictors for a model with any given amount of predictors. We looked at three different model criteria to compare models of different amounts of predictor variables: Mallow's Cp, BIC, and $R^2_{\text{adj}}$.



Although each of the three model criteria suggested a different number of predictor variables, they were all within the same region (between 6 and 8 predictors). Since they were so close, we adhered to the Occam's Razor and went with the simplest, six-variable model. This was the model favored by BIC, which gives the greatest penalty for complexity in models. The six variables used in the regression context are then `volatile.acidity`, `chlorides`, `total.sulfur.dioxide`, `pH`, `sulphates`, and `alcohol`. All of the variables from the classification selection appear here with the exception of density. In the regression context, `pH` was the variable that proved to be a rather unintuitive inclusion, since it is similar to density in the sense that it is not easily detected by the five human senses. The inclusion of these two variables in the model portray the importance of the chemical properties of a bottle of wine when it comes to quality.

## Multiple Linear Regression

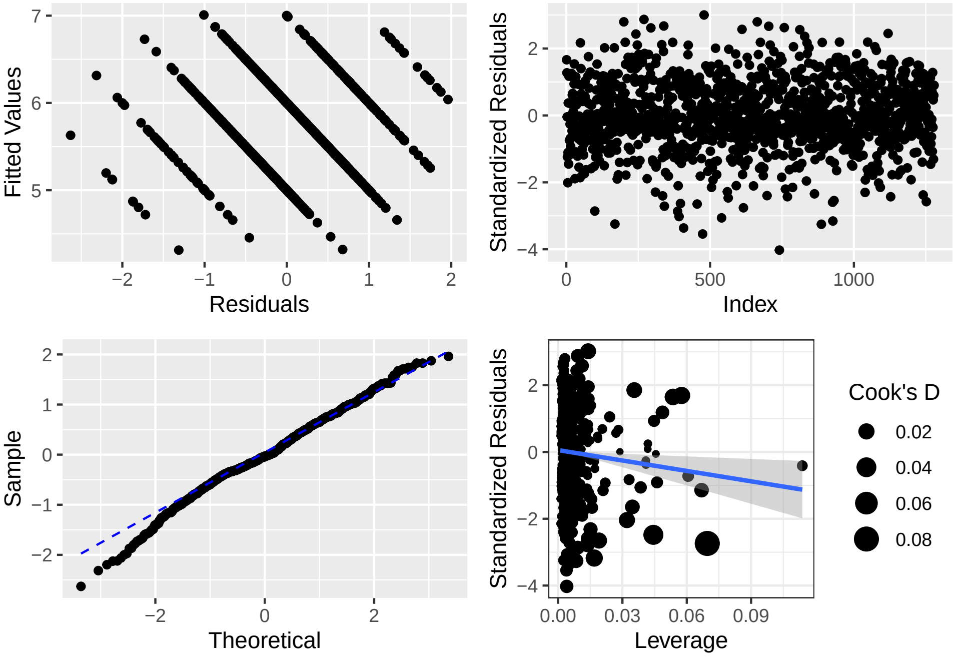
Multiple Linear Regression is the extension of simple linear regression to a set of multiple predictor variables. The goal of multiple linear regression is to model the linear relationship between a continuous response and two or more predictor variables. Like simple linear regression, coefficient estimates are found by minimizing the sum of the squared errors. The formula for a multiple linear regression model with $p$ predictor variables is as follows:

$$\begin{aligned}
Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{ip} + \epsilon_i,\\
&\text{where for } i = n \text{ observations},\\
&y_i = \text{dependent variable (response)}\\
&x_{ji} = \text{indpendent variables } (j = 1, ..., p)\\
&\beta_0 = \text{y-intercept}\\
&\beta_j = \text{slope coefficients for each variable}\\
&\epsilon_i = \text{residuals (error term of model)}
\end{aligned}$$

The multiple linear regression model is subject to the following assumptions about the data:

- Linear relationship between response and predictors
- No collinearity between predictor variables
- $y_i$'s are iid
- $\epsilon_i \sim N(0, \sigma^2)$

To assess these assumptions, we looked at various diagnostics plots:

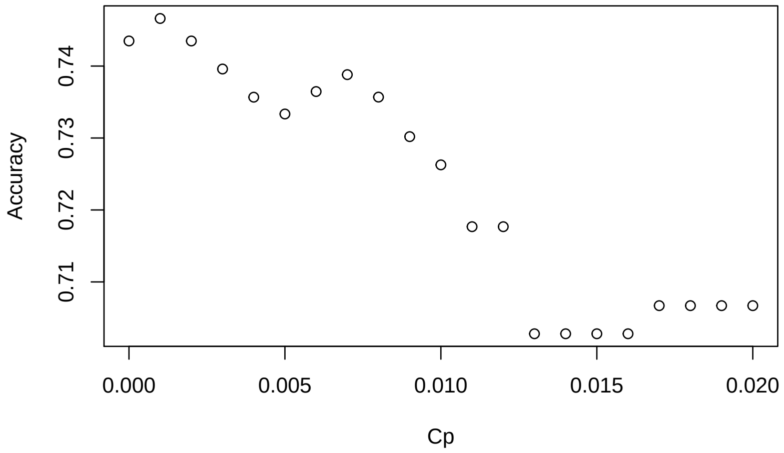


~ talk about residual plots here ~

The biggest benefit to using a multiple linear regression model is its interpretability and ease of use. The coefficients can be computed efficiently and they are have clear and easy interpretations. It is also possible to make inference on any of the variables or linear combination of variables. The most signficiant drawback of using the multiple linear regression model is that it is subject to distributional assumptions, meaning that if the assumptions are violated, our estimates and inference will be unstable at best and invalid at worst.

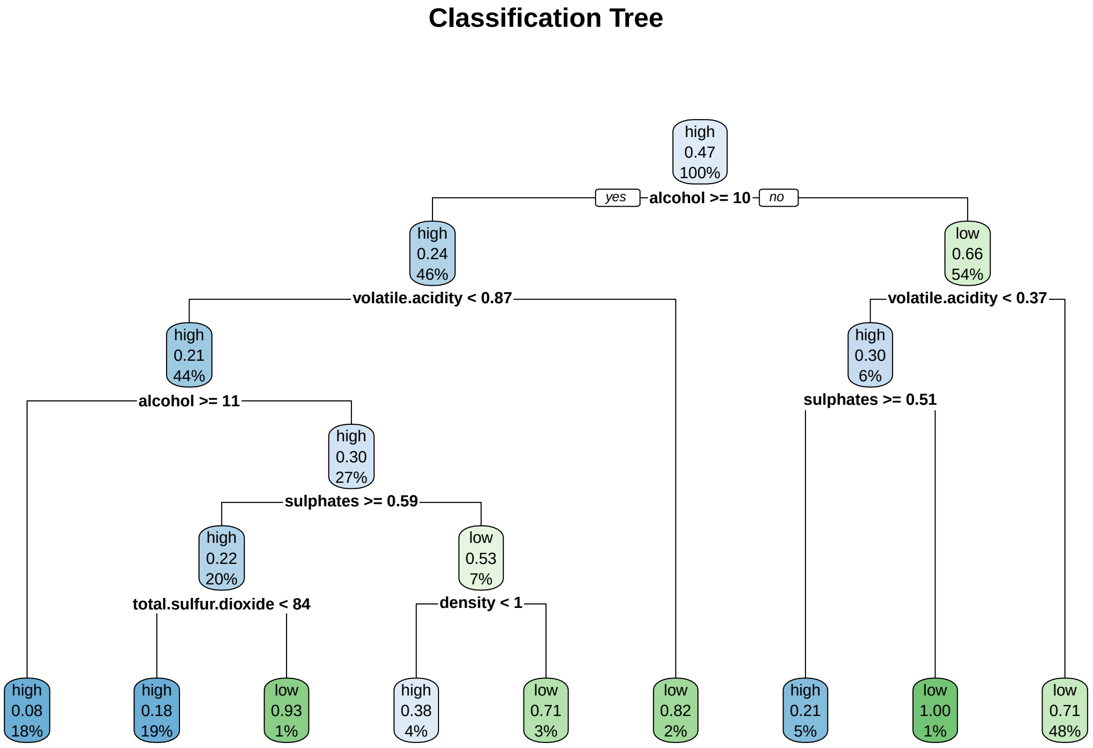
## Classification Tree

The basic classification tree is based on partitioning the data into subgroups using simple binary splitting. Initially, all objects are considered a single group. Then, the group is repeatedly binarily split into two subgroups based on the criteria of a certain variable. In the classification setting, we classify the observations in a specific region with majority vote. We want to grow the tree as big as we can, and then prune it back using cost-complexity pruning. This is done to not overfit the data, but pruning increases the bias. The pruning parameter needs to be tuned; below is a plot of the classification accuracy by pruning parameter, where .001 yields the highest accuracy.



The advantage of using a basic tree is that it is easy to understand and has a good interpretability, which is not something that translates over to ensemble methods. Additionally, the classification tree is not very computationally expensive, unlike the random forest model. We used the `caret` and `rpart` packages in R to fit our trees.

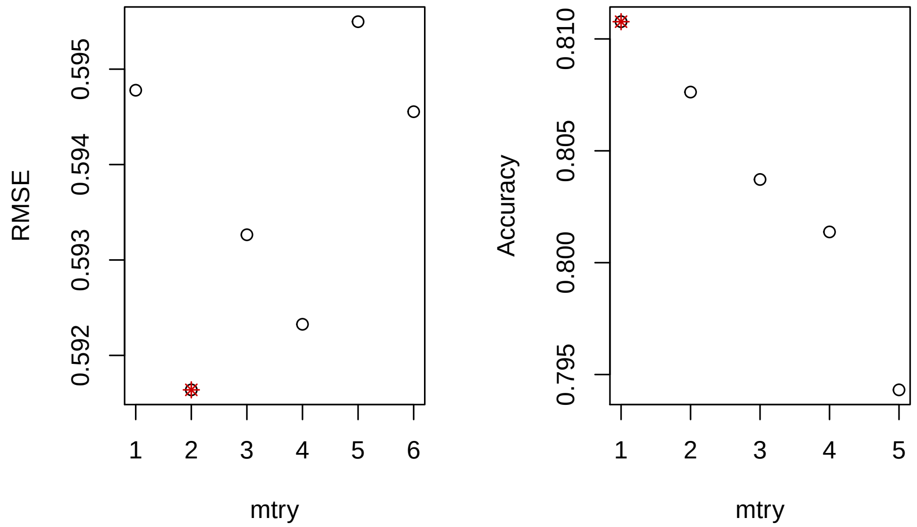
A visualization of the classification tree can be found below.



## Random Forest

The random forest model is an ensemble tree-based method, which creates multiple trees from bootstrap samples and averages the results. Many bootstrap samples are created with replacement and then a classification tree is fitted on each bootstrap sample with a random subset of the predictors. Once a prediction has been made by all of the bootstrapped trees, the final classification is based on majority vote of the bootstrap predictions. Similarly, the prediction in the regression context is an average of all of the predictions of the bootstrapped trees.

The following parameters need to be tuned in a random forest model:
* `mtry`: the number of randomly-selected variables selected at a node split * `ntree`: the number of trees to grow



Since `ntree` will plateau with enough trees, we tuned the `mtry` in both the regression and classification settings. For regression, the optimal `mtry` value is 2, while for classification it is 1. Keep in mind that the random forest is equivalent to bagging when the number of randomly selected features is equal to the total number of variables in the model. In both of the above plots, this is represented by the maximum displayed value of `mtry`.

The advantage for using an ensemble method over a regular classification or regression tree is that because there are many bootstrap samples being averaged together, there is less variance. This is similar to how the variance of the sample mean goes down as the sample size increases. Although it will probably increase our prediction accuracy, the random forest loses the interpretability that the basic trees have. Furthermore, the algorithm that is used to fit the random forest is very computationally expensive.

# Results

Table 2: Regression Model Results

| **Regression Method** | **Root MSE** |
|---|---:|
| Multiple Linear Regression (MLR) | 0.6280 |
| Random Forest | 0.5574 |

Table 3: Classification Model Results

| **Classification Method** | **Accuracy** | **Sensitivity** | **Specificity** |
|---|---:|---:|---:|
| Classification Tree | 0.6719 | 0.7241 | 0.6096 |
| Random Forest | 0.8000 | 0.8276 | 0.7671 |

Based on the output, the best model for both classification and regression was the random forest. This was not a surprise, since the ensemble methods tend to outperform regular methods. As an average of many classification trees, ensemble methods generally improve prediction accuracy and reduce overall variance. We suspect that other ensemble methods, such as bagging or boosting, may also be able to yield higher prediction accuracies.

One of the main disadvantages to using ensemble techniques like the random forest is the fact that they tend to be very computationally intensive. The basic classification tree and multiple linear regression model took significantly less time to run than the random forest, but they were less accurate in their predictions. If we wanted to run an ensemble method on a very large data set with a lot of features, it would be a computational nightmare.

# Conclusion

With the recent push for quantitative solutions in industries like agriculture, there has been an emphasis on using machine learning to solve a variety of problems requiring prediction and inference. In particular, the viticulture industry stands to gain a lot from such methods, since historically it has not interacted much with the world of analytics. In this study, we went through various machine learning techniques, assessing prediction accuracy in both classification and regression contexts.

Based on the results of our study, we were able to predict the high or low quality of a given wine with about 70-80% accuracy. Although this is a *good* result considering the non-information rate of 53%, it is by no means excellent. These results indicate that machine learning algorithms could be a good technique when trying to predict wine quality, but that there is room for improvement. With a more formal, industry expert-guided variable selection process, we believe that higher prediction accuracies can be achieved. Machine learning models could potentially be combined with other techniques to add value to winemakers.

## Limitations

The multiple linear regression model is subject to certain distributional and model assumptions, which were not explicitly tested or checked. Since the objective of this project was to predict rather than make inference, we didn't explicitly check the model assumptions.

All of the wines in the dataset come from wineries in Portugal, and although they are from different wineries, we worry that the results may not be generalizable to wine production globally. Similarly, will these prediction results hold when it comes to white wine, rose, or sparkling wine? Given this variability over location and type of wine, what can we say about the winemaking industry as a whole?

Some of the wine regions have more strict standards for the designations of wine quality. For example, regions like Bordeaux and Champagne in France have some of the highest standards in the world for their wines, which are much different than somewhere like Greece or New Zealand. How accurate will are predictions be when there are different standards and how might we account for that variation?

## Further Research

Since this study solely looks at the prediction accuracy of machine learning techniques for red wines, the natural extension of the analysis would to verify that the results hold constant for white wines as well. The University of California, Irvine Machine Learning database has data on white wines, so that would be a good place to start.

The characteristics of any given wine are often derived from different geological and/or climatological conditions. For example, a wine that comes from a region close enough to the water to get a sea breeze may have a hint of saltiness in its aroma or taste. Similarly, a grape coming from a relatively young grove might give the wine different features than if the grove were hundreds of years old. It would be interesting to see how different weather conditions, terroirs, plant ages, and slopes affect the outcome of the wine in terms of quality.