Assignment 4 – Decision Trees and Random Forests

Zach T. Adair

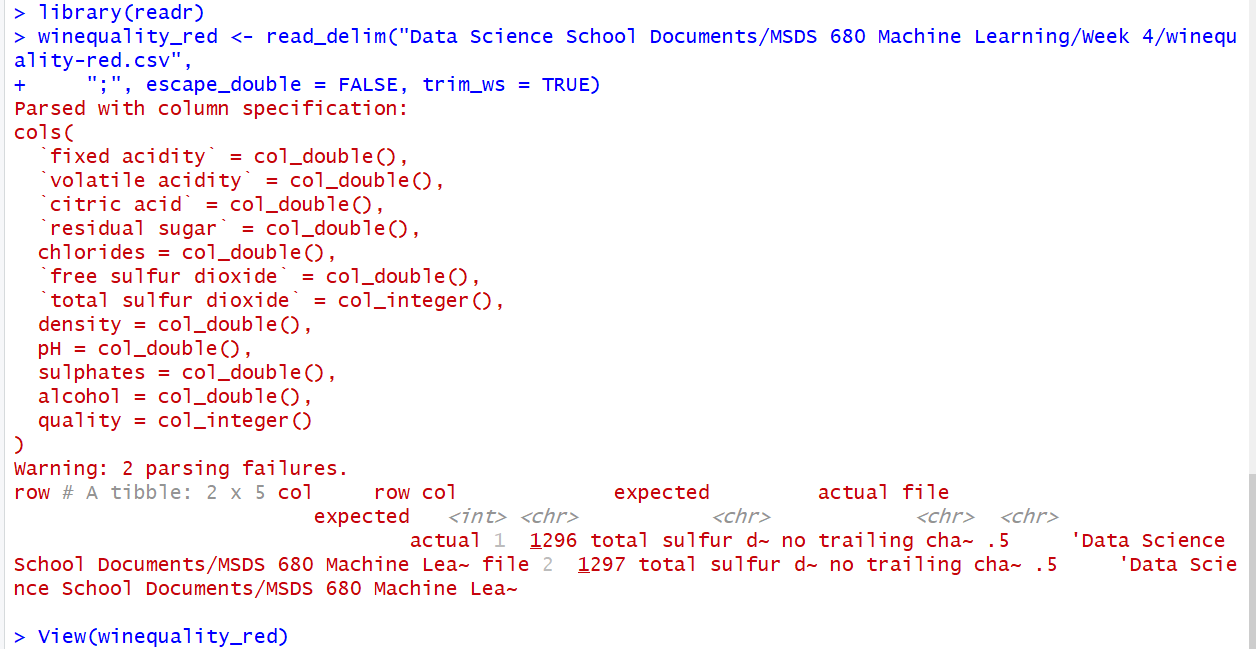
Regis University

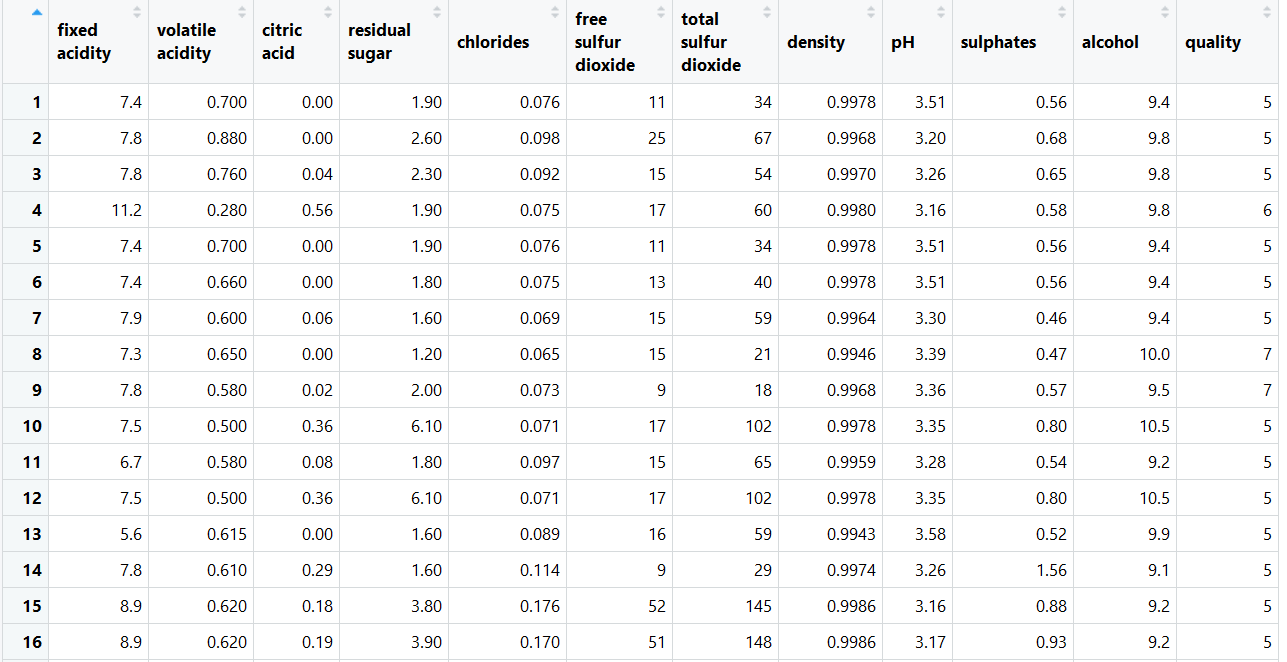
Abstract

For this week’s assignment, I will be using the red wine data provided and constructing a decision tree and random forest to help determine the quality ranking of red wines based on their chemical properties. This exercise will allow the vineyards to know what to charge, and measure quality without having to get a slew of expensive testers to determine the amount the would charge on their wine. I will uploading the data into RStudio and running some exploratory analytics upon the data. I will follow that up by preparing and creating the model based on the decision tree. Next I will test the model and decide to either prune my model or set up a random forest algorithm which is a group of decision trees. I will test that new model and see how effective my results were after that. I will finish my assignment write up with a summary detailing the process and what I have learned.

Assignment 4 – Decision Trees and Random Forests

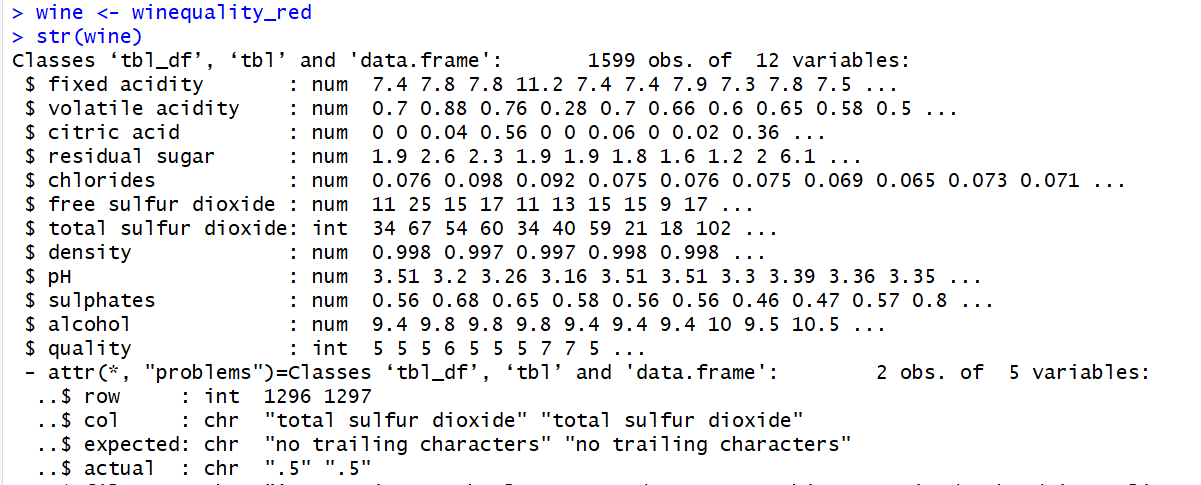
To start any assignment, the data must be loaded into RStudio and then it needs to be explored for its properties and potential trends amongst the data. So to start I loaded in the data and looked at a view of it just to see what I was working with.

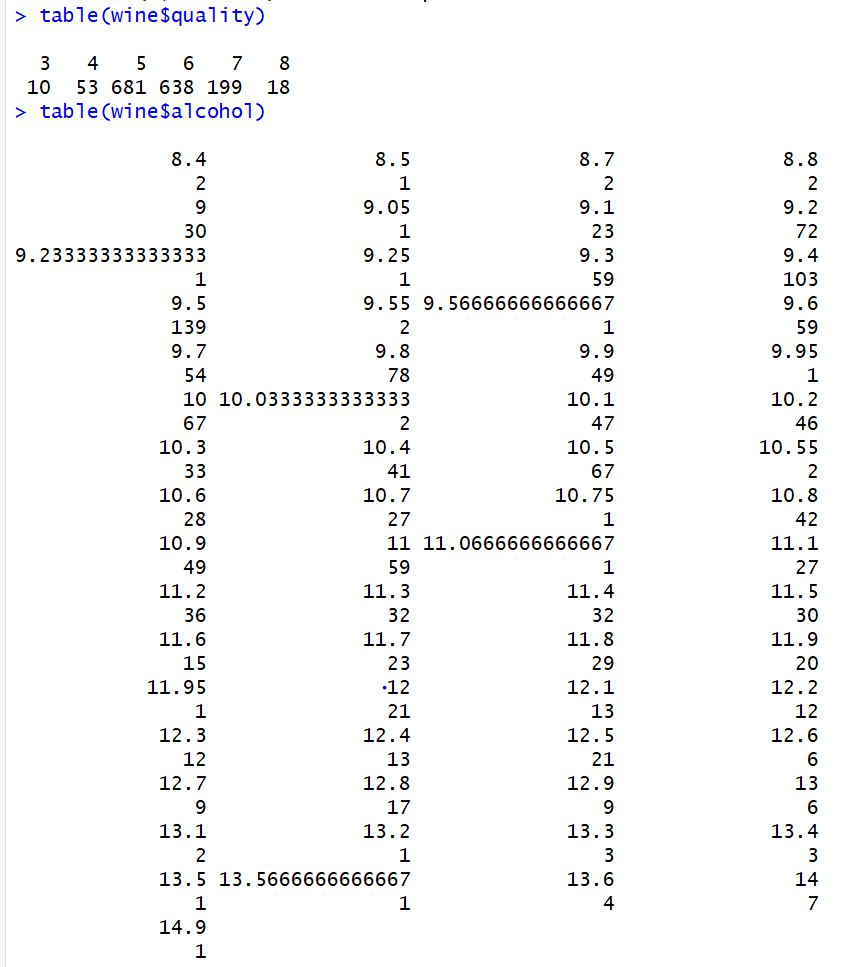


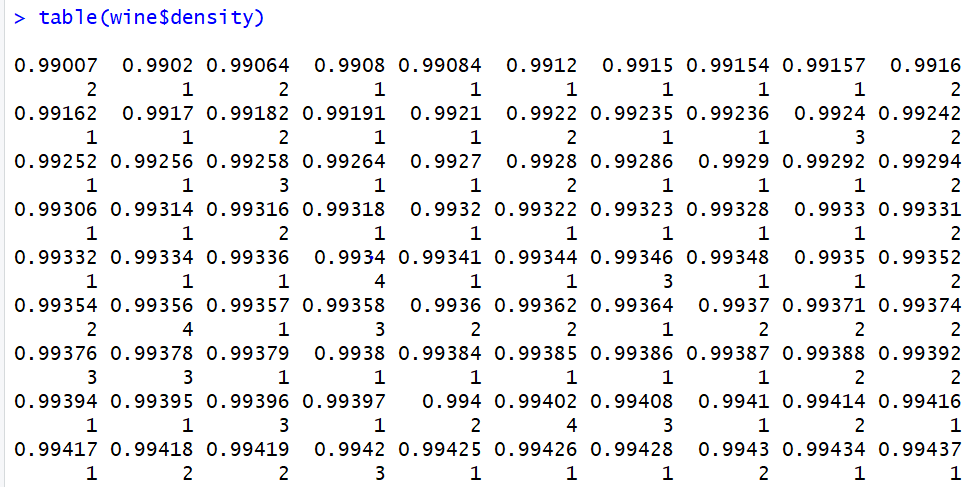


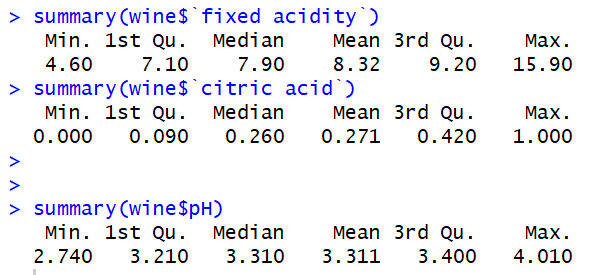
# Data Exploration

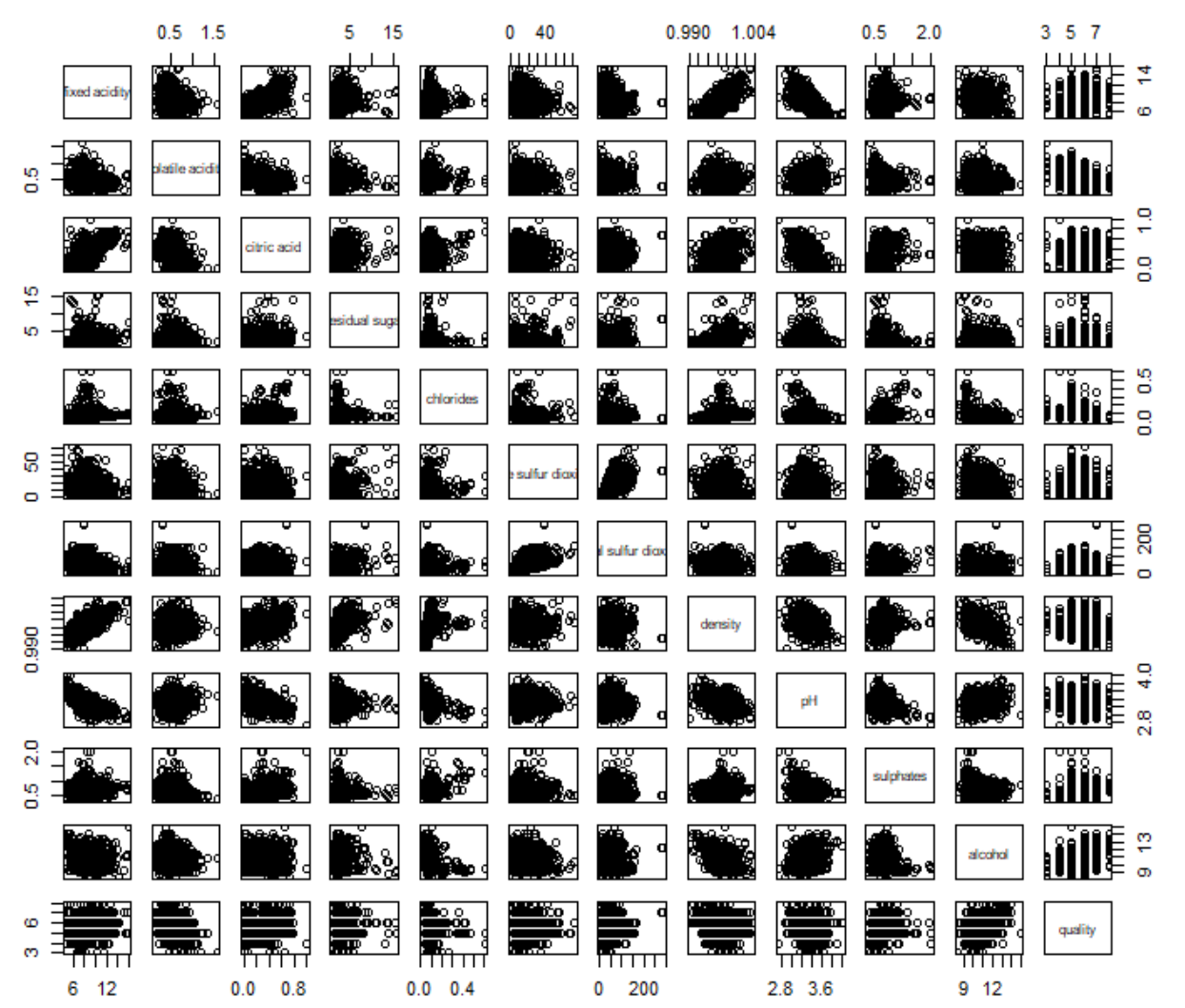
The data exploration is always an integral part of the analysis of the data, it helps the analyst see what data types they will be working with as well as see the distribution and potential trends the data might have within it. Before I got started though, I shortened the name of the data set to just “wine” then got started with my exploration of the data.

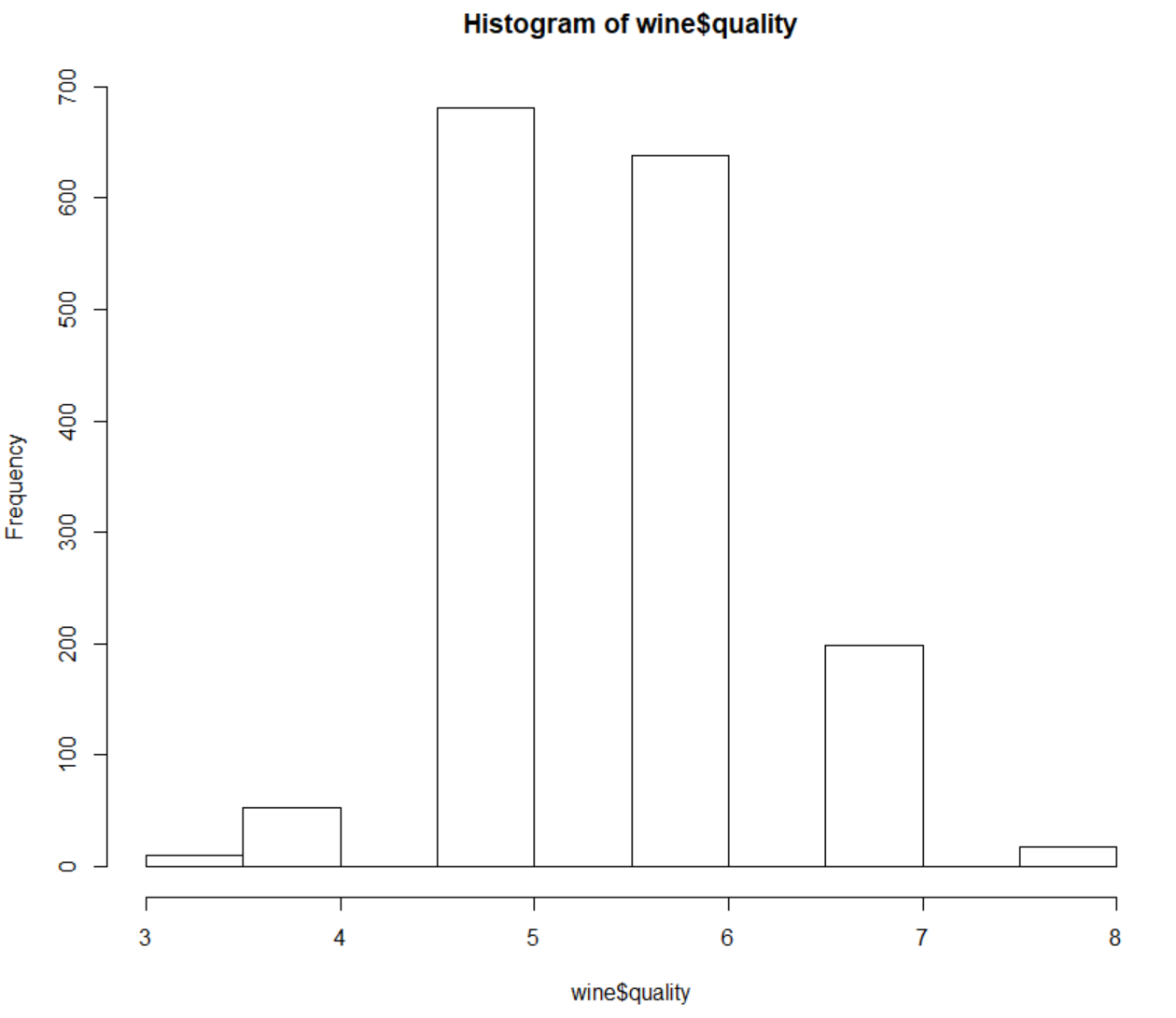








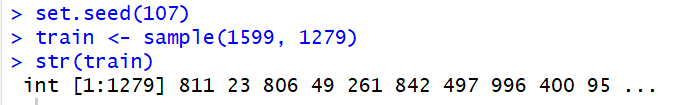




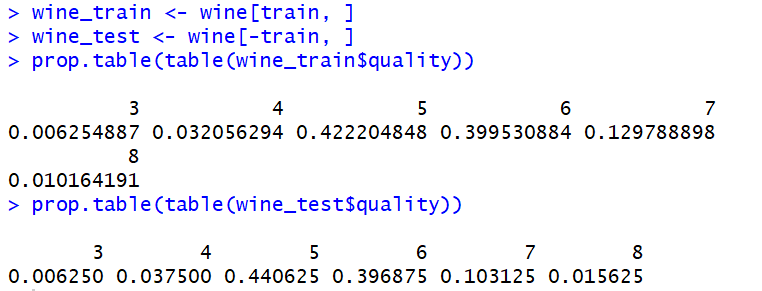
From our bit of data exploration we were able to pull some interesting details about our data. First, we know from the structure of the data set that we have a total of almost 1600 types of wine with 12 different variables of measure, which could make for a very extensive decision tree. Creating a table and looking at the summary of several variables also lead to some interesting facts about the distributions to some of the variables. Like how the quality variable has only about 5 or 6 entries within its distribution but variables like alcohol, or density have a very wide variety. Then looking at some of the chemical variables of measure the range is diverse but the max and min typically aren’t too spread out.

## Data Preparation

The data preparation portion is arguably the most important piece of the process and around 80% of the work, if done properly it leads way to the creation of great models which are great at predicting outcomes, our goal here is to make sure we prepare the data to output the models we need for good machine learning work. I need to start by splitting my data into training and testing sets. I will also need to split these by some distribution, and the one I choose is an 80:20 split. I will also set a random seed at the start, then I will create my training sample and look at its structure of random entries.



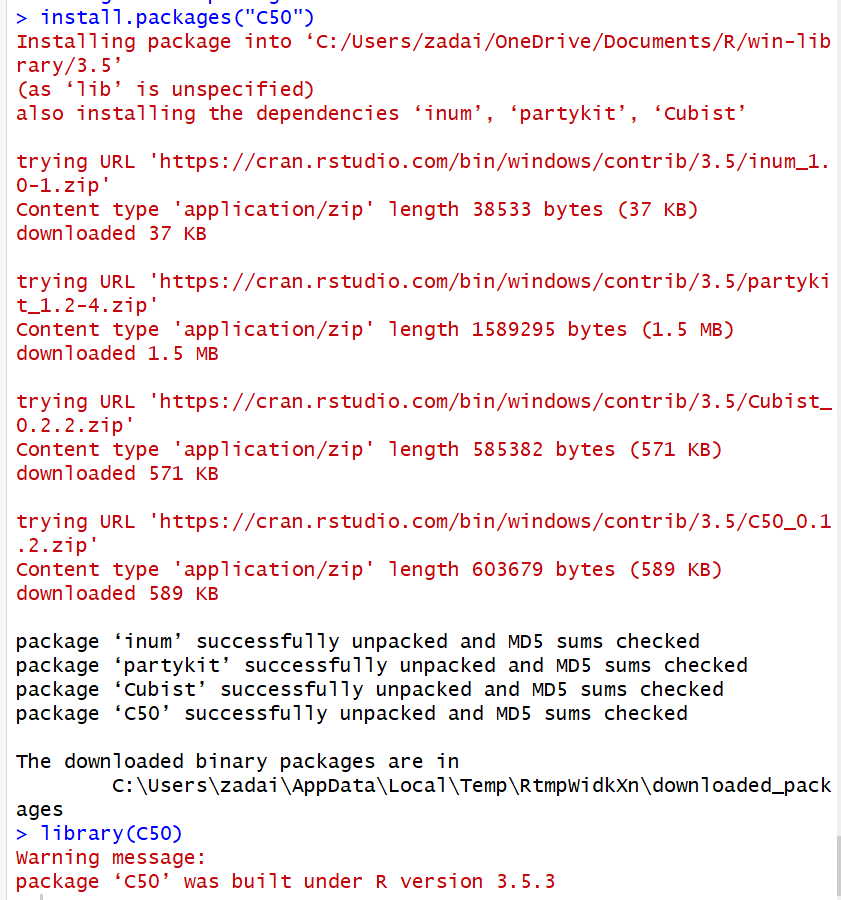
With the train sample created it is time to set up the training and test samples for the wine data.



The splits appear to be pretty similar which is what is ideal for our next step, which is training the model on the data.

### **Training the Model to the Data**

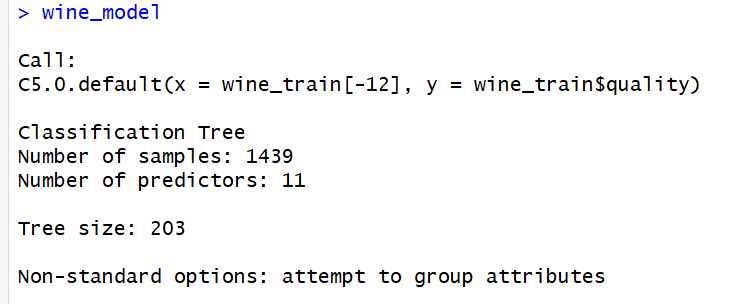
Include a period at the end of a run-in heading. Note that you can include consecutive paragraphs with their own headings, where appropriate. To start the training for this model, I will first need to bring in the C50 package, this will help with training the tree model.

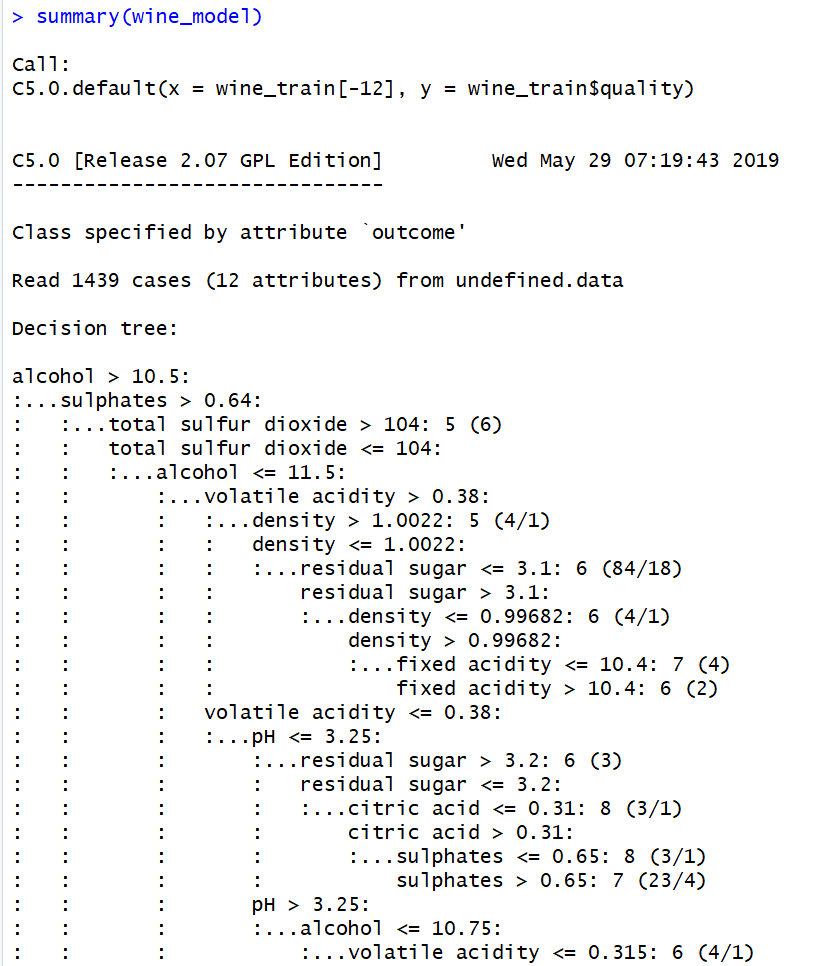


With the C50 package now installed the model can be created. Start with constructing the model.

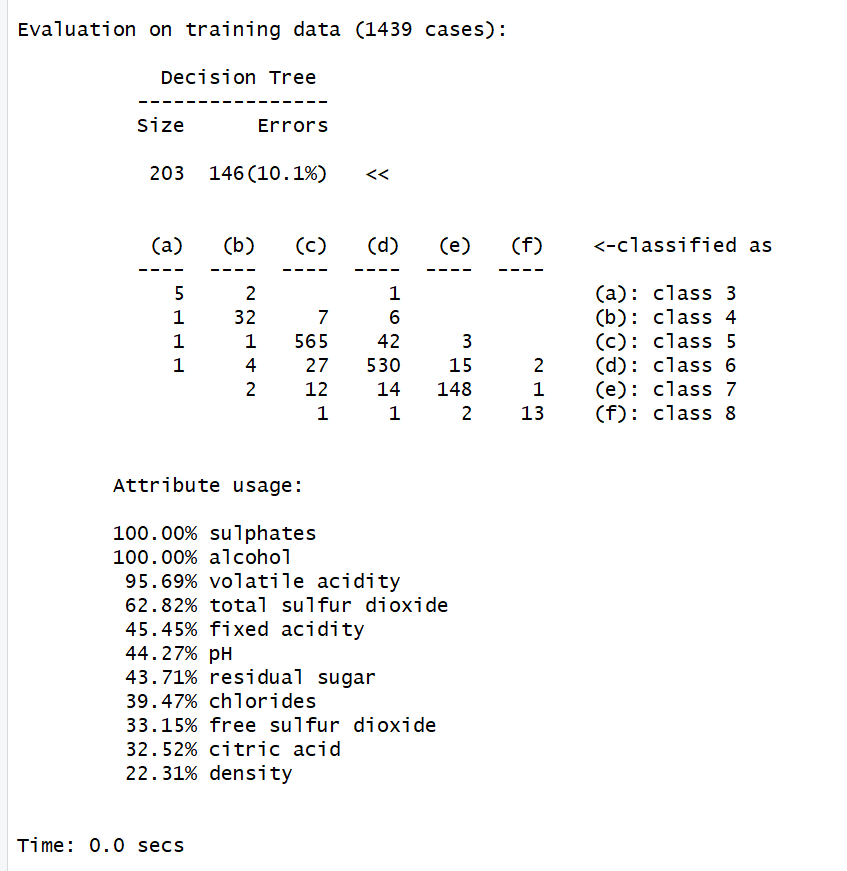


You will notice the [-12], that is the number of variables in the dataset. This helps fit the model properly to all the decisions it will have to make.





Listed out when you summarize the model is each variable according to each branch output, at the end of the summary the output spits out a confusion matrix with and a percentage of attributes used in each training data record of wine.



You can also see the decision tree has a size of 203, with 146 errors which compute to just over 10% error rate. With decision trees known for overfitting, it is safe to assume the training data could be overly optimistic. We will not no for certain though until we test the model for performance.

#### ***Evaluating the Performance of the Model***

##### Before I evaluate the performance of this model by creating a predictor model, I will first bring in the gmodels package so I can use it later for testing.

##### 

##### Now I will create the predictor model for the wine dataset, then test that model using a cross table evaluation between the actual and predicted values.

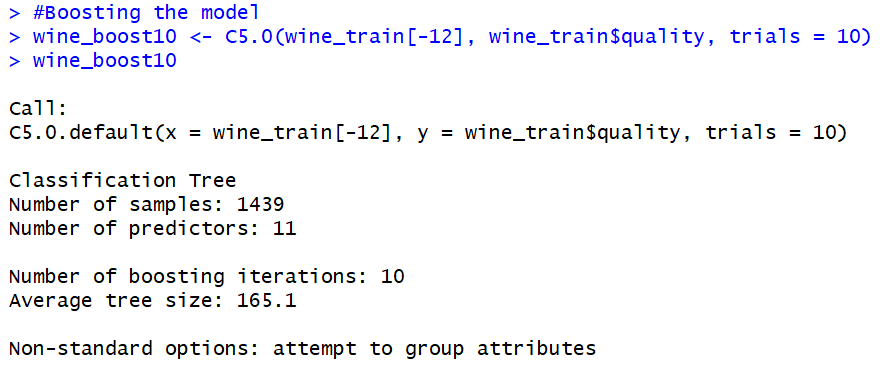
##### 

##### 

From our data we are able to validate the performance of the model based on how they line up in the table. At quality level 3, there should have been 2 values but the predictor missed both; at quality level 4, there should have been 7 values, 2 were predicted correctly and 5 were wrong; quality level 5 had 69 actual values, 51 of them were predicted correctly, the other 18 were not; quality level 6 had 59 values, 38 were predicted correctly, and the 21 others were wrong; quality level 7 had 22 values, and 12 were predicted correctly, the other 10 were wrong; and finally quality level 8 had only one value and it was predicted incorrectly. So as we can deduce from the model’s performance, it didn’t do a very good job of fitting to the trained data, but there are ways to improve a decision trees performance.

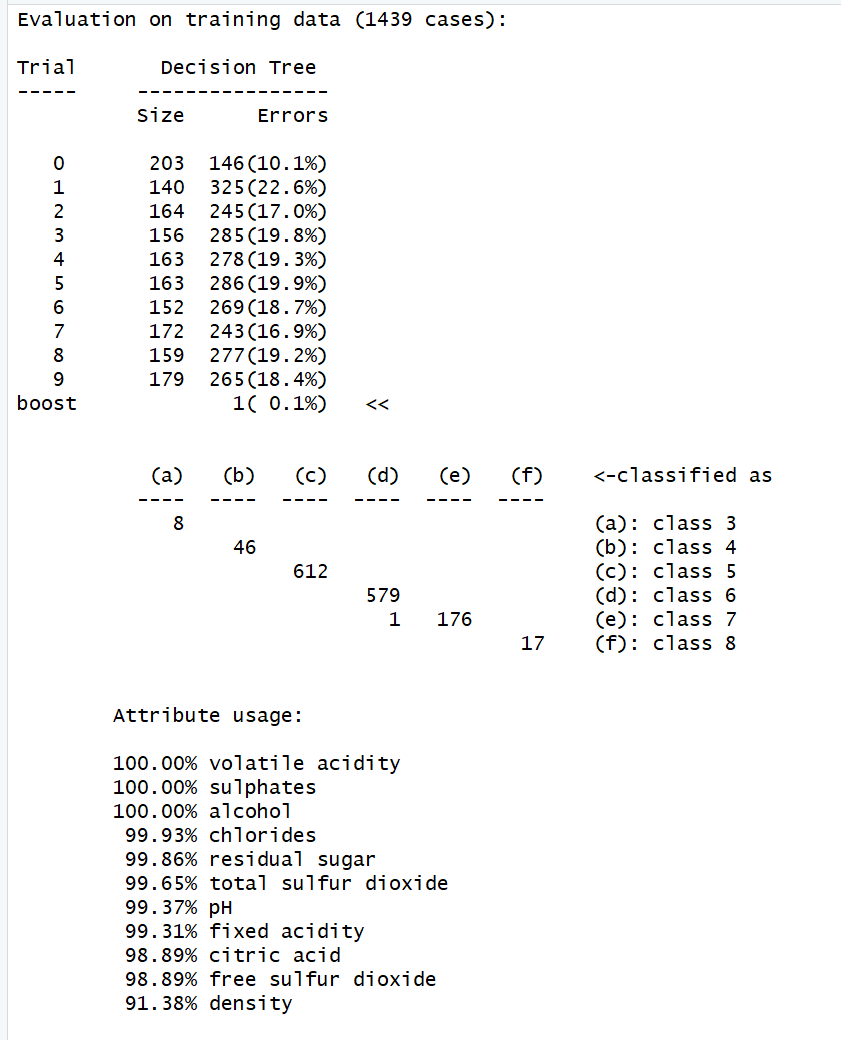
##### Improving the model

The goal in this next step will be to improve the model, which performed at roughly 50% accuracy over all the different levels of quality. This next step I will be going through the process of boosting the model, which is essentially taking several of the weaker learners, combining them together to make a stronger variable. Doing this can combine some variables with complimentary strengths and weaknesses to help dramatically improve performance. I will start by adding 10 trials to the performance of the model, from there we could see if there is any improvement in the model or if the output is similar.

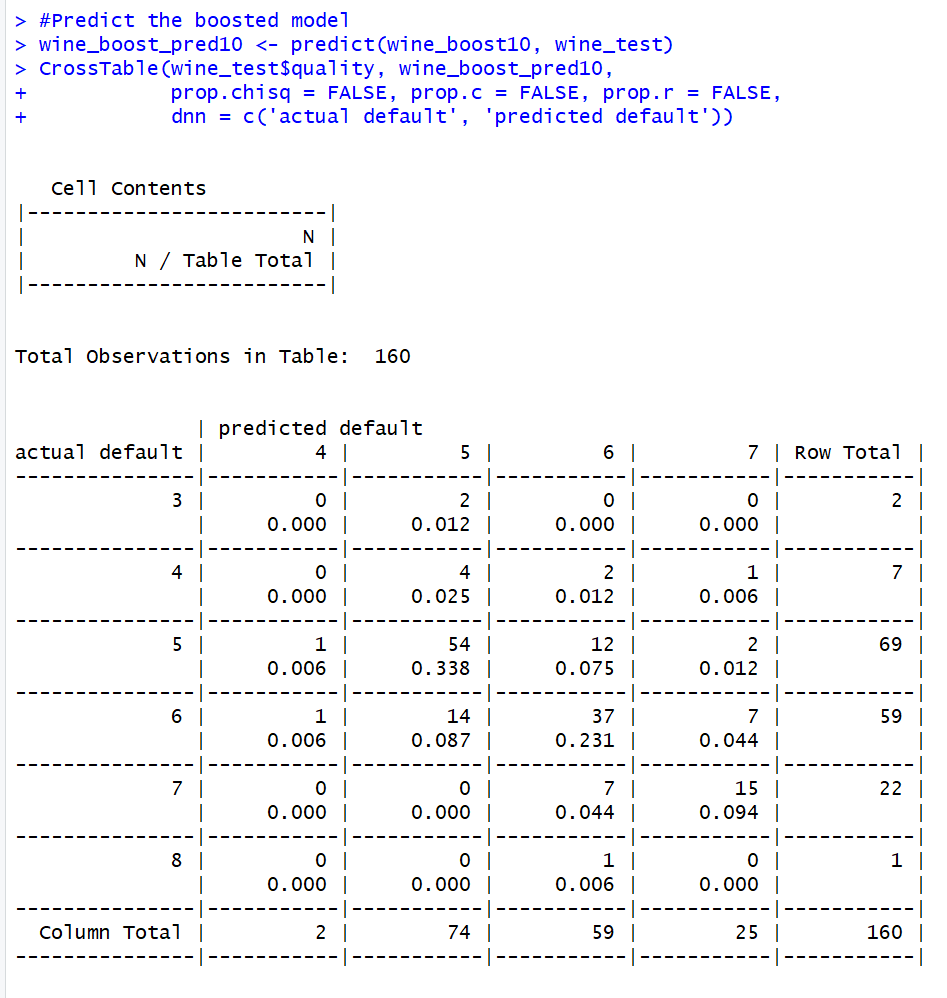


Now I will run a summary on the boosted model and check the results of the summary.





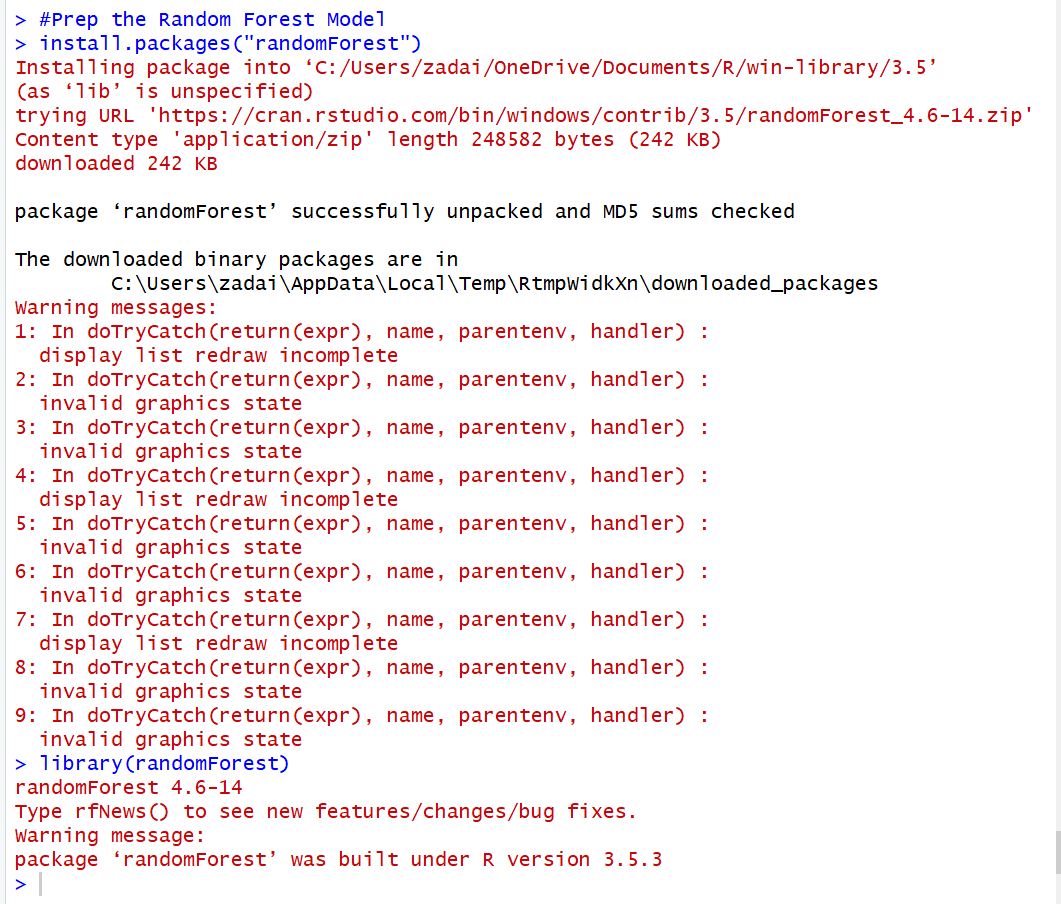
From the tail end of the summary alone we see some promise within our boosted model, the trials are divvied out pretty evenly which is good to see and the distribution of classes is similar to what we should expect with class 5 and 6 having the vast majority and then trailing off on the ends. Then the attribute usage appears to be higher for one variables who were on the lower end of the last summary of the original model. Now it is time to predict the newly boosted model and hopefully the performance is better.

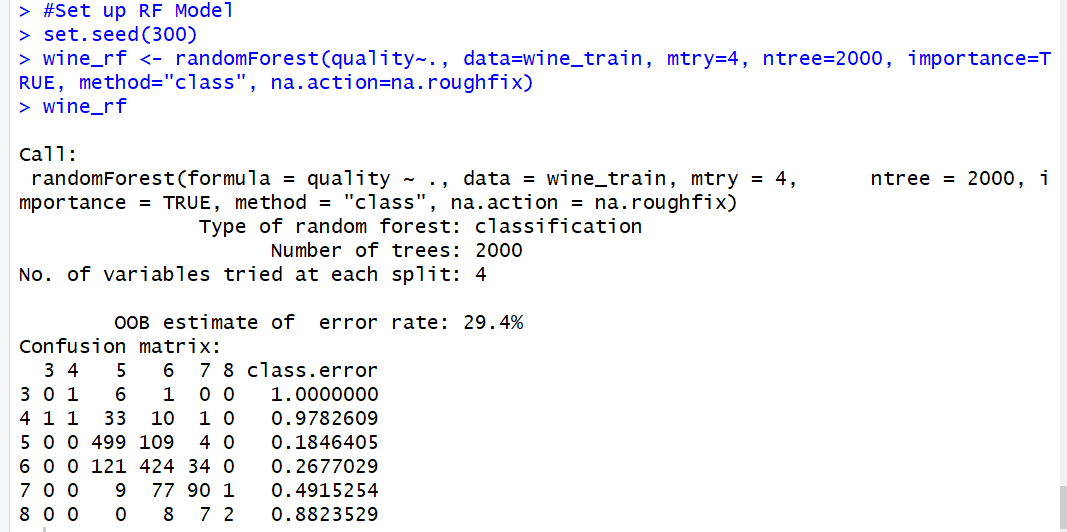


The predicted model appears to have performed a little better this time around, some things I noticed that were strange though was that the predicted default model didn’t even predict a class 3 wine through any of the test values but there were 2 and both were predicted as class 5. The overall performance of the model was 106/160 = 66.25% which is a little better than the last time but not something I would accept for a predictor algorithm, nevertheless, it was worth the effort to try and get the model to improve. I want to improve the performance of my model and since I’ve tried boosting to little effect I want to try another method, the method that I want to try next is Random Forests.

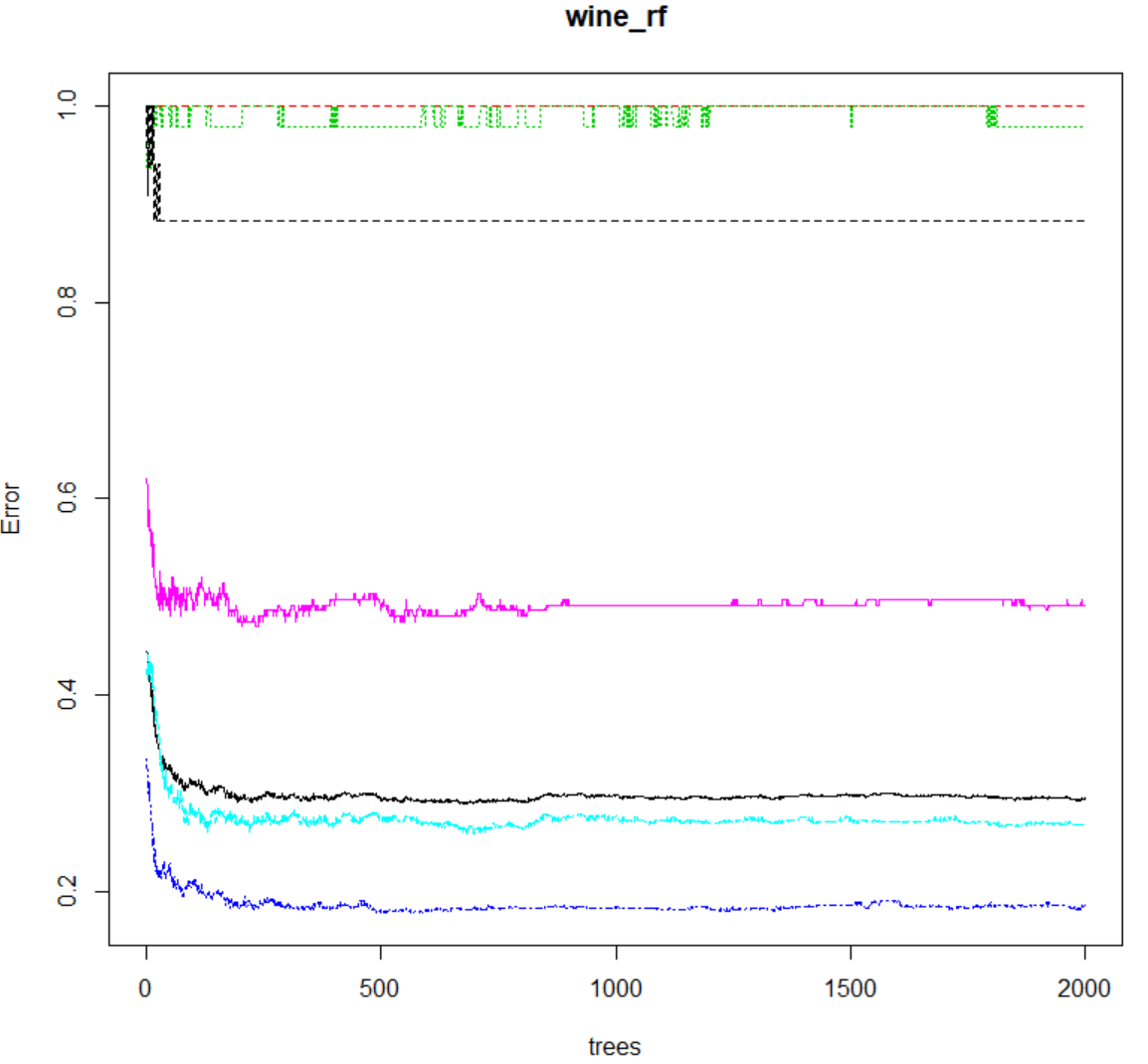
**Random Forests**

Random Forests are a great option to use to enhance a Decision Tree model. The characteristics of a Random Forest are that they are competitive by nature and prone to not overfitting which is terrific because Decision Tree models have the tendency to be prone to overfitting. The one thing a Decision Tree has over a Random Forest model is that it is harder to interpret the results of a Decision Tree. To put my data through this model I will need to bring in the random forest package to RStudio.

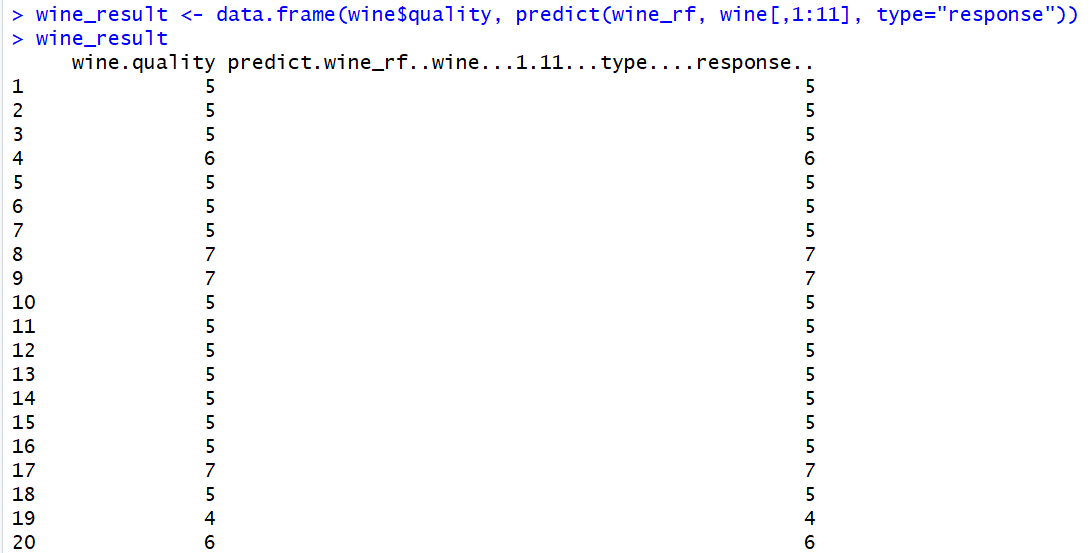


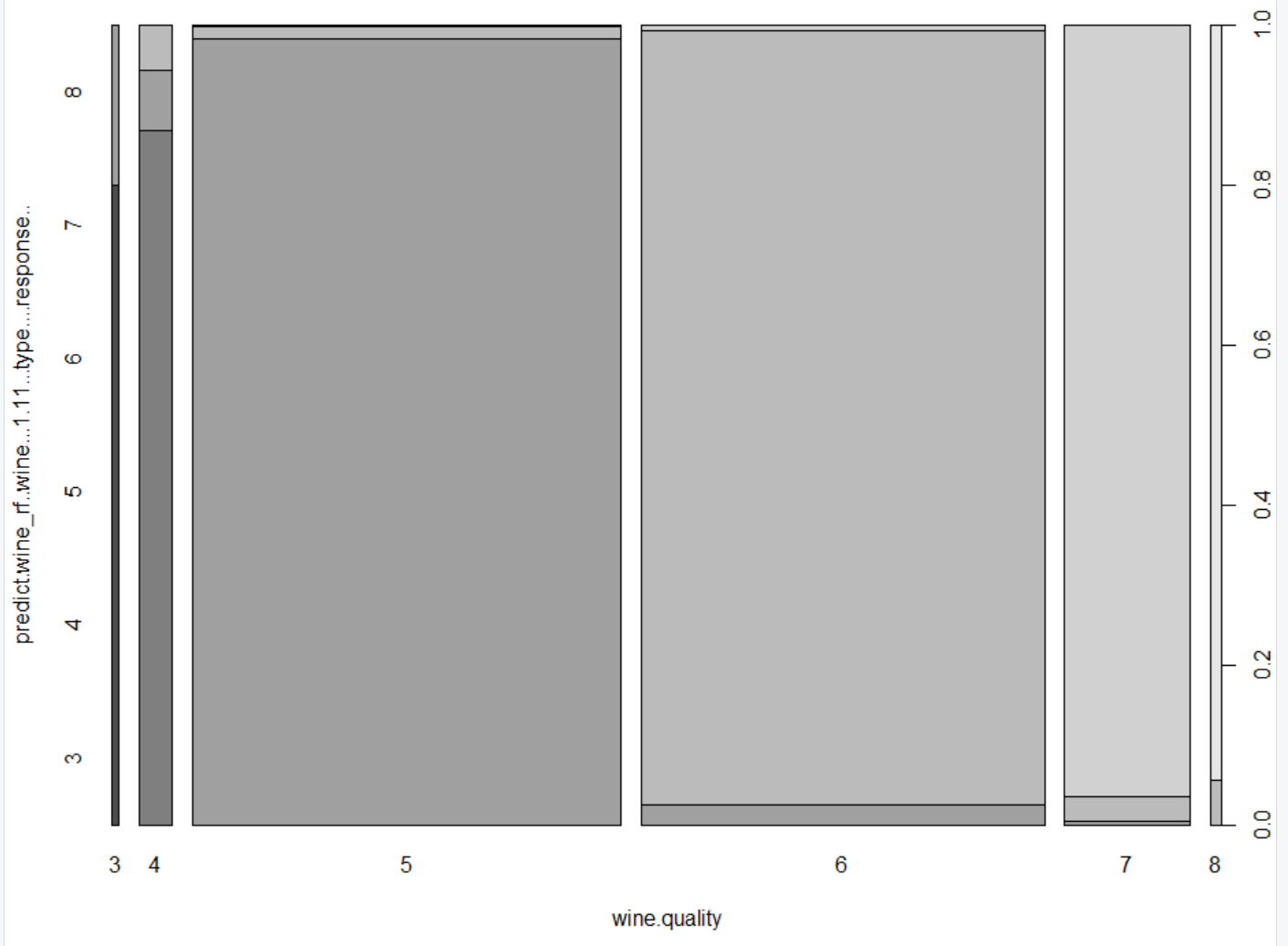






The Random Forest model has now been set up and the confusion matrix for the model has been output. I set the seed this time to 300 just to try and get the value as close to the 80% mark as I could. With the model set up its now time to evaluate the Random Forest model.





With the result now plotted and looked at we can see based on how the Random Forest predicted, it wasn’t to high, just over 70% prediction accuracy but that was still pretty decent compared to our earlier models and isn’t bad something that is just pertaining to wine. But our vast majority of class 5 and 6 wines did predict very well and so did our class 4 it would see from the plot above.

**Summary**

From this assignment I got to learn about applying Decision Trees to data in RStudio, I then was able to take it to the next level and try to improve the model with a technique called boosting, than I tried again by using a Random Forest model. Decision Trees and Random Forests are interesting concepts in Machine Learning because they are quite simple and can be easy to explain to people who are not as knowledgeable about other Machine Learning concepts. The one drawback that I can see happening a lot with these kinds of models is that they can be very easy to either under or over fit the model to the data so it takes a couple cycles of readjusting to get a data model which fits the data properly.

References

Kuhn, M. (2010). Variable Selection Using The caret Package, Pages 1- 21.

Lantz, B. (2015). Machine Learning with R (Second Edition): Packt Publishing.