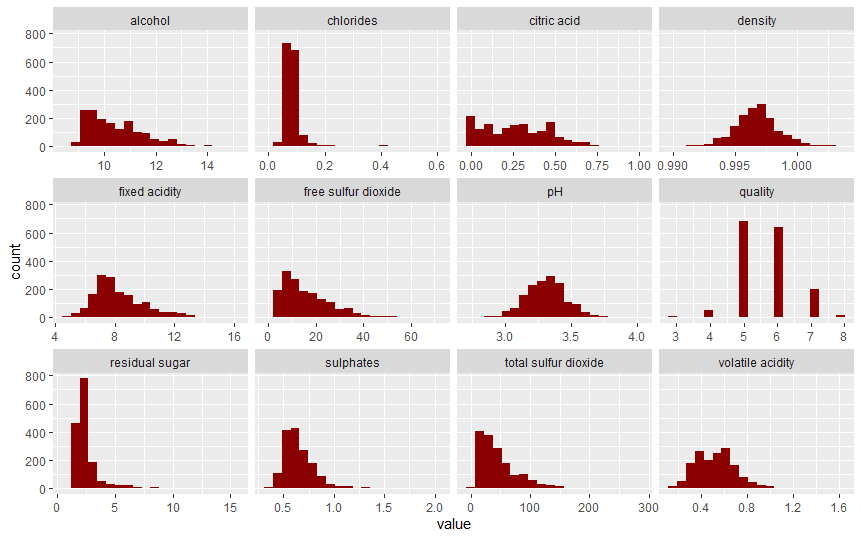
**Introduction**

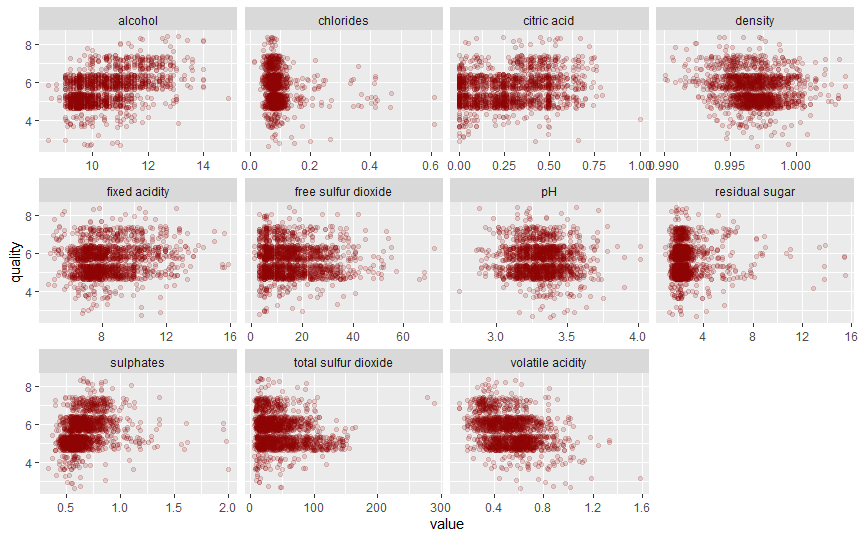
The data set we are using contains 1,599 observations of various red wines originating in the “Vinho Verde” region in the northern part of Portugal. The data set contains 11 continuous predictor variables each containing information regarding various physiochemical properties of the wine. These physiochemical properties include properties such as citric acid content, chlorides, and pH level of the wine. The response variable in the data set is the quality score of the wine. This variable is an integer value ranging from 1 to 10. For the purpose of our analysis, when utilizing regression techniques to predict quality, we will treat the response as continuous. For analysis using classification methods, we will create a binary response variable which will be 1 if the quality score is above average and 0 if it is below average.

**Exploratory Data Analysis**

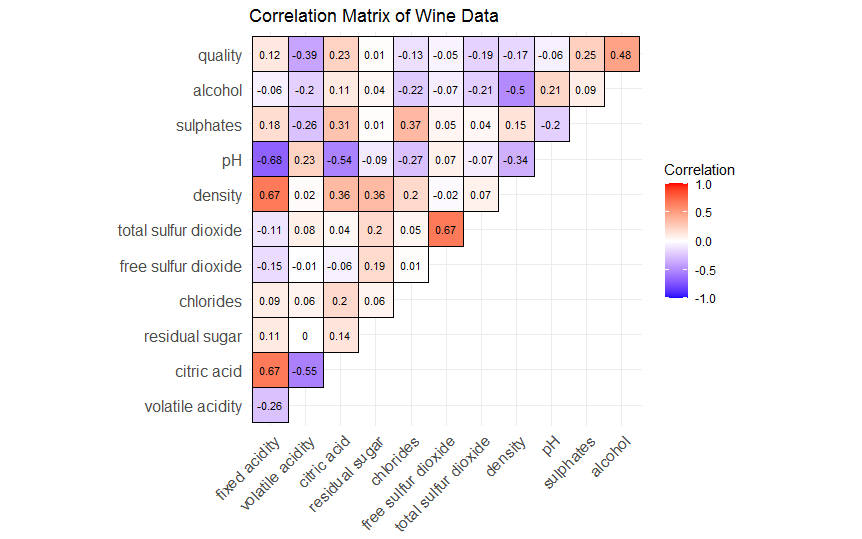
The first step in our analysis was to create some plots to understand what our data looks like. To get an understanding of how each of the variables in the data set are distributed, we created a histogram for every variable. The histograms can be seen below in **Figure 1**.

**Figure 1:** Histograms of each variable in the wine data set

From this figure, we can see that each of the variables is reasonably distributed and none will cause too much concern when it comes to model fitting. Chlorides and residual sugar appear to have the vast majority of observations concentrated around a small range of values, with a few observations that are potential outliers. We can see the lowest quality score is 3 and the highest quality score is 8. Quality is an ordinal variable, with most wines receiving a quality score of 5 or 6 and a smaller number of wines receiving very high or very low scores.

The next part of our exploratory analysis involved visualizing the relationship each variable had with the response variable, quality. **Figure 2** presents scatterplots of quality versus each predictor variable. Jitter was added to the plots and the opacity of the points was reduced to manage the overlapping of points due to quality being an ordinal variable with the vast majority of values at either 5 or 6.

**Figure 2:** Scatterplots of quality versus each predictor variable

From the scatterplots, it appears that chlorides, residual sugar, and sulphates have a few outliers, while the other points are clustered in a small range of values. It appears that alcohol has the strongest positive correlation with quality and volatile acidity has the strongest negative relationship. This is confirmed by the correlation matrix visualization shown below in **Figure 3**.

**Figure 3:** Correlation matrix of wine data

The correlation matrix above provides a little more clarity on the relationship each predictor variable has with quality and with each other. Alcohol has the strongest correlation with quality out of all the variables at .48. Residual sugar has the weakest at .01, which is virtually no correlation at all. There are a couple predictor variables that are relatively strongly correlated with each other, such as pH and fixed acidity at -.68 and free sulfur dioxide and total sulfur dioxide at .67. However, this is unlikely to be a major issue thanks to the machine learning algorithms that we will use.

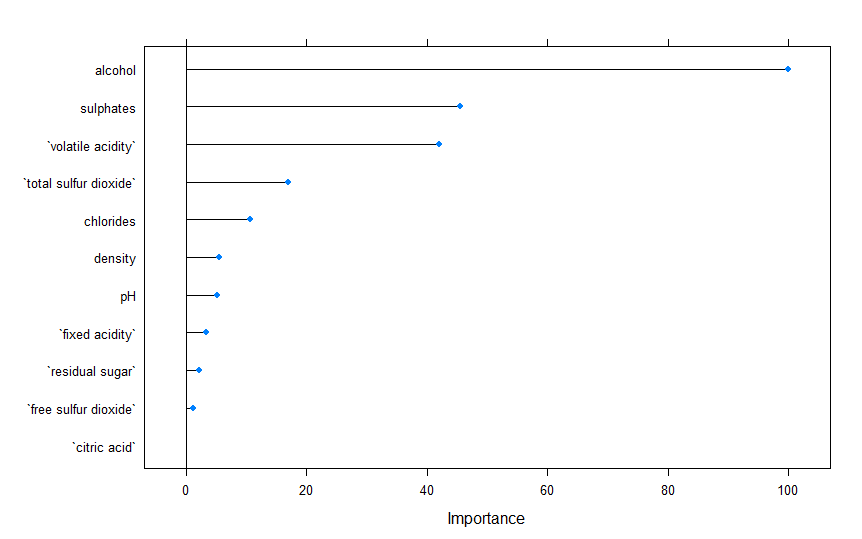
**Regression**

Some of the most effective machine learning techniques for regression tasks are ensemble methods, such as bagging, random forests, and boosting. These ensemble methods are most often used to improve predictive accuracy of tree-based methods. Individual trees often have higher variance than other regression methods and can also have more bias, but ensemble methods can improve predictive accuracy of individual trees by building hundreds or thousands of trees to eventually generate a prediction. The ensemble method we used was bagging, which stands for bootstrap aggregation. Bagging uses the bootstrap to create a larger number of training data sets and fits a model to each of these training data sets, then averages all the predictions to come up with a final prediction. By averaging over a larger number of predictions, bagging can significantly reduce the variance of a prediction, and therefore improve mean squared error (MSE). In our analysis, we used bagging to create 500 trees to predict quality using all 11 predictors. The bagged model resulted in a test MSE of .328.

Another powerful ensemble model is a random forest. In bagged tree models, it is possible for one dominant predictor variable to be the initial split in almost all trees. To avoid this problem, a random forest randomly chooses a small subset of variables that can be used when building a specific tree. This ensures that the same variable is not playing a dominant role in every tree, which can further help reduce variance. The number of variables allowed to be used is usually chosen to be the square root of the total number of variables or the number of variables divided by 2. In our analysis, we used 5-fold cross-validation to determine the number of variables to be used in the trees and ended up with 6 (out of a possible 11). Our random forest model again generated 500 trees to predict quality. The random forest model resulted in a test MSE of .322.

The final ensemble method that we used in our analysis is called gradient boosting. Gradient boosting is similar to bagging and random forests in that it grows multiple trees to generate a prediction, but how those trees are built differs significantly. Where bagging and random forests grow a large number of *separate* trees, gradient boosting uses the results from the first tree to grow the second, the results from the second to grow the third, and so on. Specifically, it first sets the prediction (call this f-hat) to be 0 and residuals for each observation to be the values of the response. It then fits a tree using the residuals as the response, generates a prediction, and adds a “small portion” of that prediction to f-hat. It then updates the residuals and repeats the process until the total number of trees have been grown. F-hat at this time is the final prediction. The “small portion” of the prediction I mentioned earlier is called the learning rate, and this parameter is usually chosen via cross-validation. In our analysis, we used 5-fold cross-validation to select the number of trees and the learning rate. The selected model used 1000 trees and a learning rate of .01. The gradient boosted model had a test MSE of .355.

One interesting aspect of ensemble methods is the aspect of variable importance. Ensemble methods grow a large number of trees and each tree has various splits within it. In regression, each split reduces the MSE by a certain amount. When we grow numerous trees, for each variable we can determine the average amount the MSE decreased due to splits on that variable. Variables that decrease MSE the most are considered to be the most important variables and vice versa. We can plot the importance of each variable to see the relative importance of each variable. **Figure 4** shows an example of a variable importance plot.

**Figure 4:** Variable importance plot for our random forest model

In the plot above, the importance was scaled to have a max of 100. We can see alcohol is far and away the most important variable. Sulphates and volatile acidity come in second and third, respectively. Importance plots for the other models had the same top three variables.

..

..

Other regression methods

..

..

|  |  |
| --- | --- |
| **Model** | **Test MSE** |
| Bagged Trees | .328 |
| Random Forest | .322 |
| Gradient Boosted Trees | .355 |

**Table 1:** Test MSE of regression methods

**Classification**

The ensemble methods used in the regression setting can also generate very accurate predictions for classification. Due to the nature of the task at hand, prediction for classification is much different. Whereas in regression bagging and random forests generate predictions by averaging the predictions of all of the trees, in classification a prediction is made by taking a majority vote, i.e for a given set of features, the class that is predicted most often by the trees is chosen. Classification predictions for gradient boosting are also much different but have not been outlined our text and are therefore excluded here as well. In the classification for our analysis, 500 trees were fit for the bagged model and the random forest. 5-fold cross-validation was again used to select the tuning parameters for the random forest and the gradient boosting classifier. For the random forest, the number of predictors in each tree was selected to be 4. For the gradient boosting classifier, the number of trees was again chosen to be 1000, and the learning rate was again chosen to be .01. Our bagging model was able to generate test prediction accuracy of 84.21%. The random forest was able to generate test prediction accuracy of 83.21%. The gradient boosting model was able to generate test prediction accuracy of 80.7%.

|  |  |
| --- | --- |
| **Model** | **Test Accuracy** |
| Bagged Trees | 84.21% |
| Random Forest | 83.21% |
| Gradient Boosted Trees | 80.7% |

**Table 2:** Test accuracy of classification methods