Predicting wine quality using classification models

[didemch]

2024-02-08

Introduction

After taking a Wine Studies course, I became interested in Winemaking. Winemaking is a complex craft as wine flavor and quality depend on many different factors such as acids, alcohol compounds, pH of the grape juice and others. I am interested in finding out whether it is possible to predict the quality of the wine because it would make it easier to find a good wine for laypeople.

Each wine has many different flavors; after drinking the same wine, some people may think it was fruity and some people may find it too sour. Hence, each person will judge the wine based on their own perception. But I thought it would be fun and interesting to find out whether we could "predict" the wine quality in advance based on its chemical features. Then, there might be a universal "grading" scheme for wines based on facts (values) rather than on people's wine tasting skills and preferences.

In this project I want to find out what would be the best classification model for the wine quality prediction. Along with that, I am interested in determining factors that influence the quality of the wine the most. I also want to find out whether there are any interactions between the factors themselves or whether there are any confounding factors? (Factors that are not considered but may influence wine quality significantly). After building classification models, I want to find prediction errors (such as CV or OOB) and determine any prediction risks involved.

Dataset and Methods

The dataset includes various parameters of Portuguese "Vinho Verde" white wine.(Cortez et al. 2009) The data comes from the UCI Machine Learning repository as it has the most number of observations and predictors compared to other wine data sets that I came across. In the dataset there are 4898 observations with the following 11 qualitative attributes derived from physiochemical tests: `Fixed acidity`, `vol acidity`, `Citric acid`, `res sugar`, `Chlorides`, `Free sulfur dioxide`, `Total sulfur dioxide`, `Density`, `pH`, `Sulphates`, `Alcohol`. Finally, the 12th attribute is subjective `quality` out of 10.

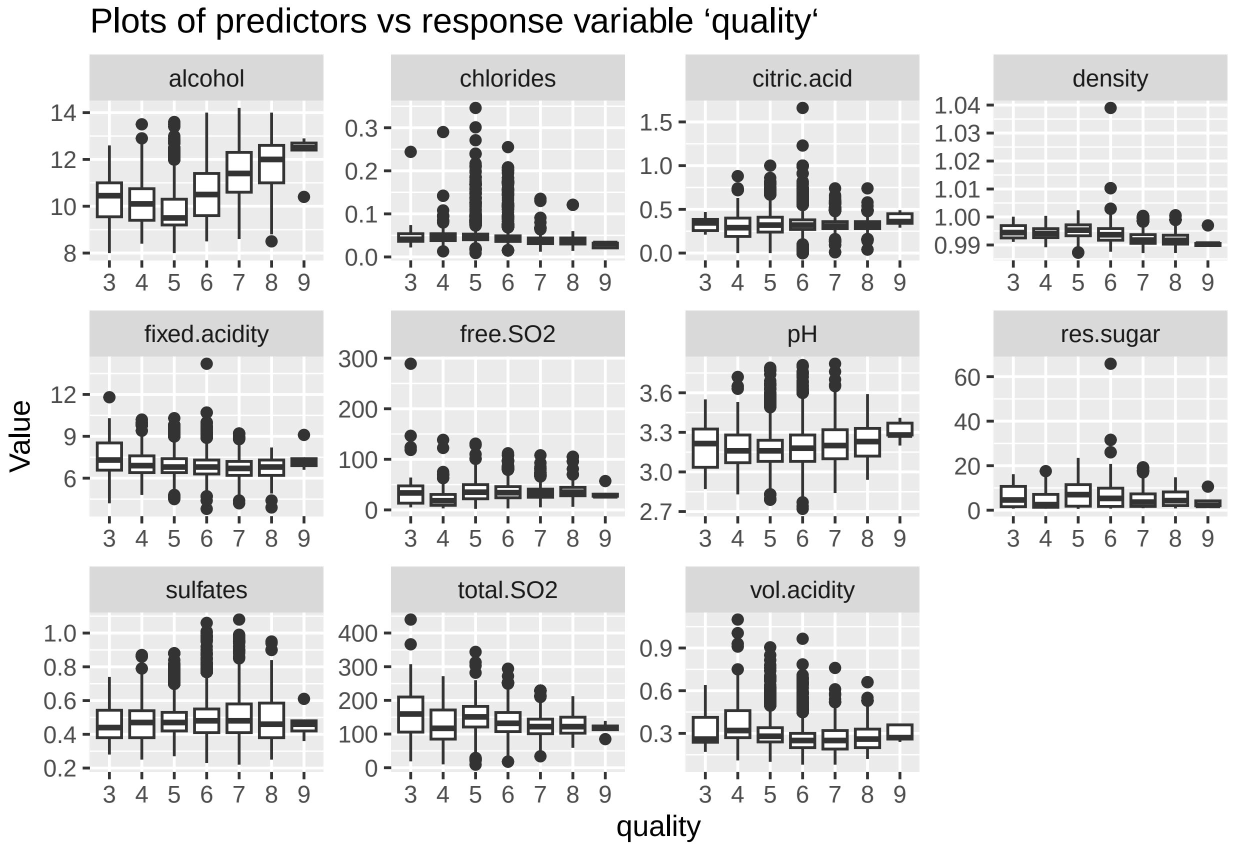
Since I am trying to create a classifier that aims to predict the quality of white wine, where the quality is on an ordinal scale from 1 to 10 (1, 2, 3, 4, 5, 6, 7, 8, 9, and 10), it is not appropriate to use linear regression because model predictions can fall outside of the score range. Instead, I will attempt using different classification methods such as Classification Trees, Random forests, or KNN on this data and see what method would result in a more accurate prediction and what features are most indicative of a good quality wine.

I begin with an exploratory data analysis in order to inspect initial relations between the predictors and response variable.

Exploratory data analysis

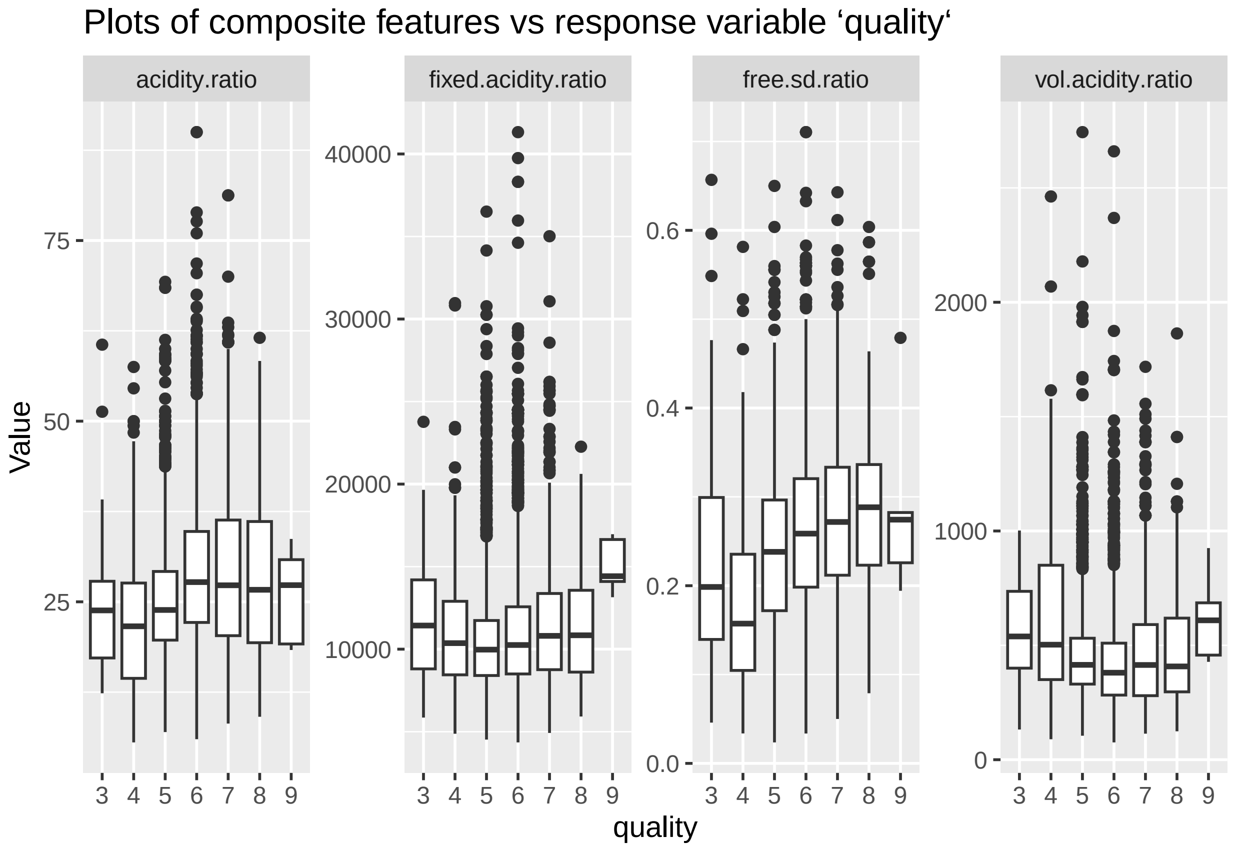
Univariate and bivariate analysis

The following plots show the 11 predictors against `quality`, the variable of interest. The purpose of these plots is to examine the relationship between each predictor and wine quality, as well as the distribution of each individual predictor.



The above plots reveal outliers in all of the measurements. However, `fixed.acidity`, `citric.acid`, `vol.acidity`, `pH` , `chlorides` and `sulfates` all show an especially high number of outliers. While there doesn’t appear to be reason to question the validity of the data, these outliers are something to consider when creating the models. These plots do not show any incredibly strong relationships. However, there is a clear non-linear positive correlation between `quality` and `alcohol`. Furthermore, `quality` appears to be negatively correlated with `chlorides`, `total.sulfur.dioxide` and `density`. Alongside the 11 predictors that came with the data, it might be valuable to explore new predictors created from existing ones. Four new predictors are generated, all relating to the ratio of related compounds. For ratios involving `pH`, it is converted back to a linear scale. The intuition behind this is that flavour is generally about balancing many components. Presumably, the ratios between various chemicals of interest might reflect whether some flavour components are “balanced” and thus correlate with the quality. The ratios are

Name	Numerator	Denominator
acidity.ratio	fixed.acididty	vol.acididty
free.sd.ratio	free.SO2	total.SO2
fixed.acidity.ratio	fixed.acidity	$10^{-\mathrm{pH}}$
vol.acidity.ratio	vol.acidity	$10^{-\mathrm{pH}}$



While it certainly isn’t a strong relationship, there does appear to be some positive correlation between `quality` and `acidity.ratio` and `free.sd.ratio`. Furthermore, it appears that `fixed.acidity.ratio` and `vol.acididty.ratio` may influence `quality` in some non-linear way however, the changes on the plot could just be random noise.

The above plots show fairly clearly that it will be hard to find a direct relationship of any value between any individual predictor and quality. However, there does appear to be some underlying relationship between some predictors and quality.

Variable Selection

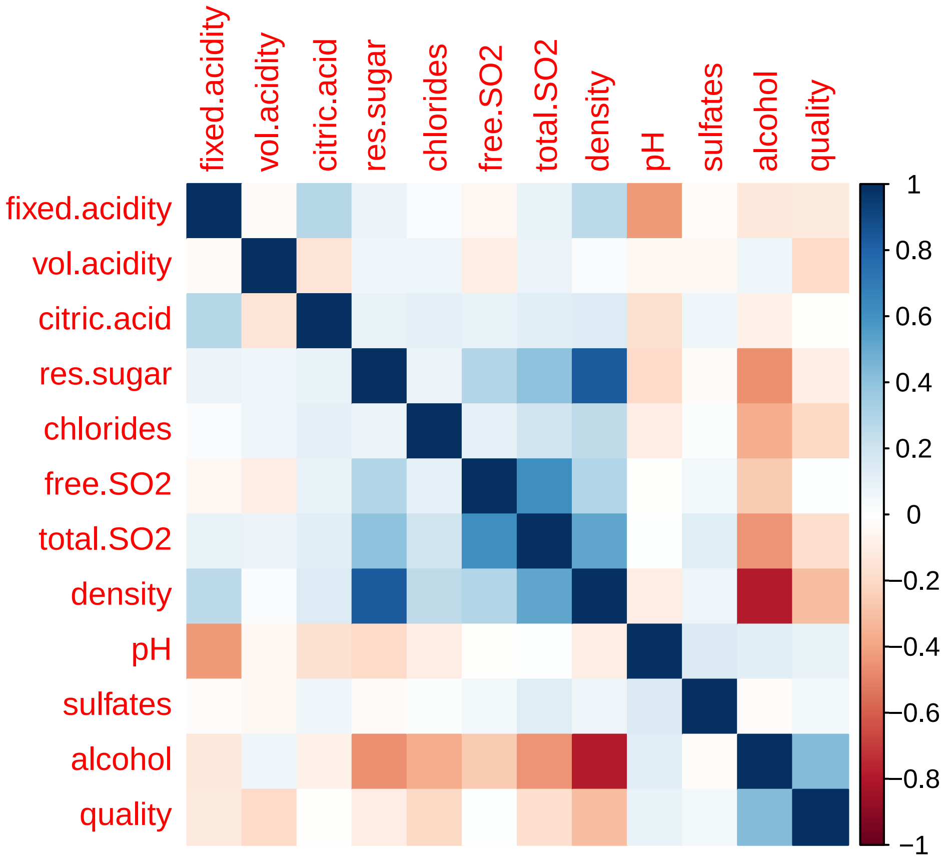
I can find out the significant factors using variable selection. Since there are only $2^{11} = 2048$ possible models, it is still computationally feasible to test all possible selections. Here is the variable selection table:

```
##           fixed.acidity vol.acidity citric.acid res.sugar chlorides free.SO2 total.SO2 density pH  sulfates alcohol
## 1  ( 1 )  " "           " "         " "         " "       " "       " "      " "       " "     " " " "      "*"    
## 2  ( 1 )  " "           "*"         " "         " "       " "       " "      " "       " "     " " " "      "*"    
## 3  ( 1 )  " "           "*"         " "         "*"       " "       " "      " "       " "     " " " "      "*"    
## 4  ( 1 )  " "           "*"         " "         "*"       " "       "*"      " "       " "     " " " "      "*"    
## 5  ( 1 )  " "           "*"         " "         "*"       " "       " "      " "       "*"     "*" " "      "*"    
## 6  ( 1 )  " "           "*"         " "         "*"       " "       " "      " "       "*"     "*" "*"      "*"    
## 7  ( 1 )  " "           "*"         " "         "*"       " "       "*"      " "       "*"     "*" "*"      "*"    
## 8  ( 1 )  "*"           "*"         " "         "*"       " "       "*"      " "       "*"     "*" "*"      "*"    
## 9  ( 1 )  "*"           "*"         " "         "*"       " "       "*"      "*"       "*"     "*" "*"      "*"    
## 10  ( 1 ) "*"           "*"         " "         "*"       "*"       "*"      "*"       "*"     "*" "*"      "*"    
## 11  ( 1 ) "*"           "*"         "*"         "*"       "*"       "*"      "*"       "*"     "*" "*"      "*"    
```

This shows the variables that needed to be included based on AIC and BIC values. In other words, it also shows the order of variables I should remove if I want a reduced model. For example, for model with 8 variables, `citric.acid`, `chlorides` and `total.SO2` should be removed from the model.

Are there any interactions between the factors?

To investigate possible interactions, I can first find the correlation matrix of the data and represent using a heat map:



Without considering the actual correlation values, it is possible to see from the heat map that there is:

- significant positive correlation between `res.sugar` and `density`
- significant negative correlation between `alcohol` and `density`

Logically, the correlation and direction make sense as I do expect if sugar content of a liquid increases, density should increase. On the other hand, as alcohol is less dense than water, I expect as alcohol content in wine increases, density of the wine decreases. These findings will prove useful when attempting to fit LDA and logistic regression models.

Analysis

The following section describes multiple approaches towards classifying the data and their results.

Normalization

Every predictor (including composite predictors) was normalized to a mean of 0 and a variance of 1 with the formula $x_i = \frac{x-\bar{x}}{SD(\bar{x})}$ for use in KNN, random forest and neural network classifiers. This normalization was used so as to eliminate the effect of predictor magnitude on model fit as different predictors had wildly different magnitudes. For example, `free.SO2` has a range in the hundreds while `chlorides` has a range of size less than 0.4. Without normalization, `free.SO2` would be hundreds of times more important than `chlorides` in the KNN model for no real reason whatsoever. Standardization around mean and variance instead of standardization around min and max (i.e. each variable is scaled to a min of 0 and a max of 1) was chosen so that outliers do not disproportionately effect the standardized values. For example, a single outlier in `res.sugar` approximately doubles the maximum which would mean that every other value of `res.sugar` under the aforementioned standardization scheme would be double if that single value weren't present.

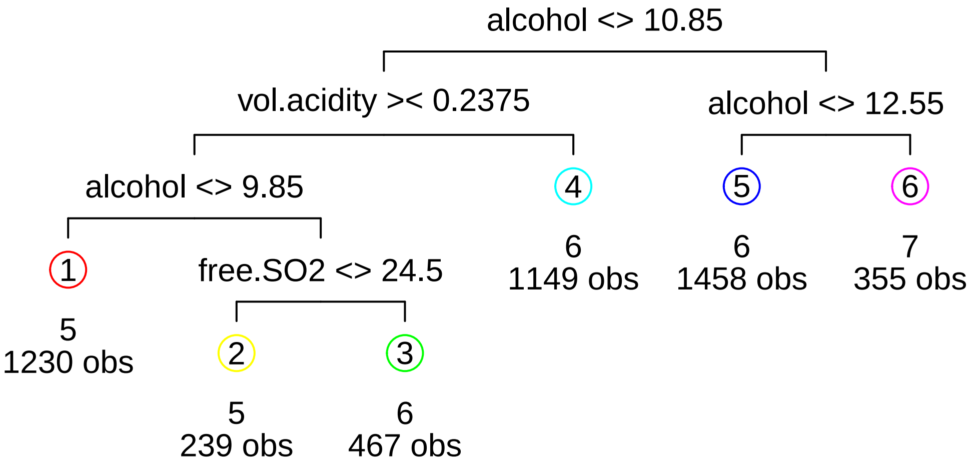
LDA & Multinomial Logistic Regression

In addition to the models below, an attempt was made to conduct Linear Discriminant Analysis (LDA) and multinomial logistic regression on the data set. In the case of LDA, I find that the out-of-sample classification error is 0.4682. This means that about 46.82 % of classifications by LDA are wrong. Thus, LDA is not a good approach to classifying this data set. As for multinomial logistic regression, the training error obtained is 0.4596 and test error is 0.4653, which means the model will misclassify about 46.53% of the time. In other words, a multinomial logistic regression model is able to correctly predict wine quality only about 53.47% of the time, which is not a reliable prediction method.

Classification Tree

Classification trees can easily be displayed and interpeted using a dendogram even when there are 11 features in our data set. Hence, I start by building a tree (classifier) that minimizes the Gini index.

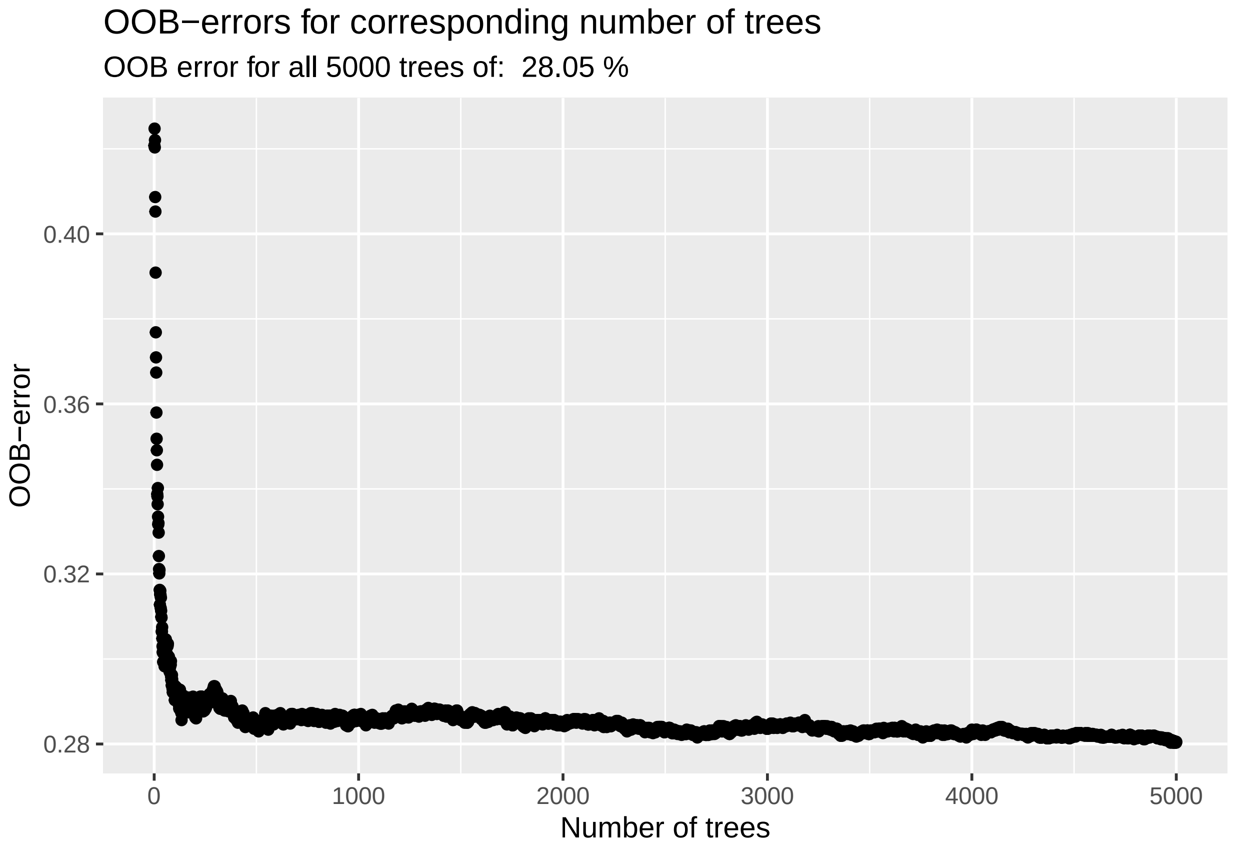
Dendogram for Classification tree:



Looking at the dendogram above, I can see that our tree classifies all the observations with `quality` of 5, 6 and 7. This means that all of the observations with wine `quality` of 3, 4, 8, and 9 get classified incorrectly. According to the dendogram, for the wine to have the `quality` of 7, it is required to have `alcohol` amount of 12.55 or more. Similarly, among wines with `vol.acidity` greater than 0.2375, all wines with `alcohol` content less than 9.85, wine will have a `quality` of 5. Hence, it is possible to make a hypothesis that the more alcohol content there is in wine, the better quality this wine is. However, it is important to state that classification trees have a high variance which implies that our conclusions could radically change with a small change in the input data. Thus, classification trees are inadequate as a classifier for this data.

Random Forest

Random forests (RF) is a nonlinear classifier technique that reduces the variance compared to trees and bagged trees by growing each tree on a random subset of predictors, thereby decorrelating the trees. Instead of using cross-validation, RF models will be evaluated based on Out-of-Bag (OOB) error which consists of predicting each observation using only the trees for which it wasn't chosen.



At first, multiple RF models with different features were fit. Incorporation of composite features did not result in a smaller OOB-error and an attempt to remove highly correlated features like `res.sugar`, `density` and `alcohol` does not seem reasonable since these features seem to be the most important when minimizing the Gini index.

Hence, an RF model with all original features and a large number (5000) of trees was chosen in order to build a plot above and to find out the number of trees that would lead to a minimized OOB-error. For 5000 trees, the OOB-error is about 28.1%. The plot shows that the estimated prediction risk nearly plateaus after 500 trees. Error does appears to continue to decrease very slowly, thus, using all 5000 trees is justified. However, it should be noted that 5000 trees takes a while to compute and thus, a smaller number of trees would be a better size if computing time was a concern.

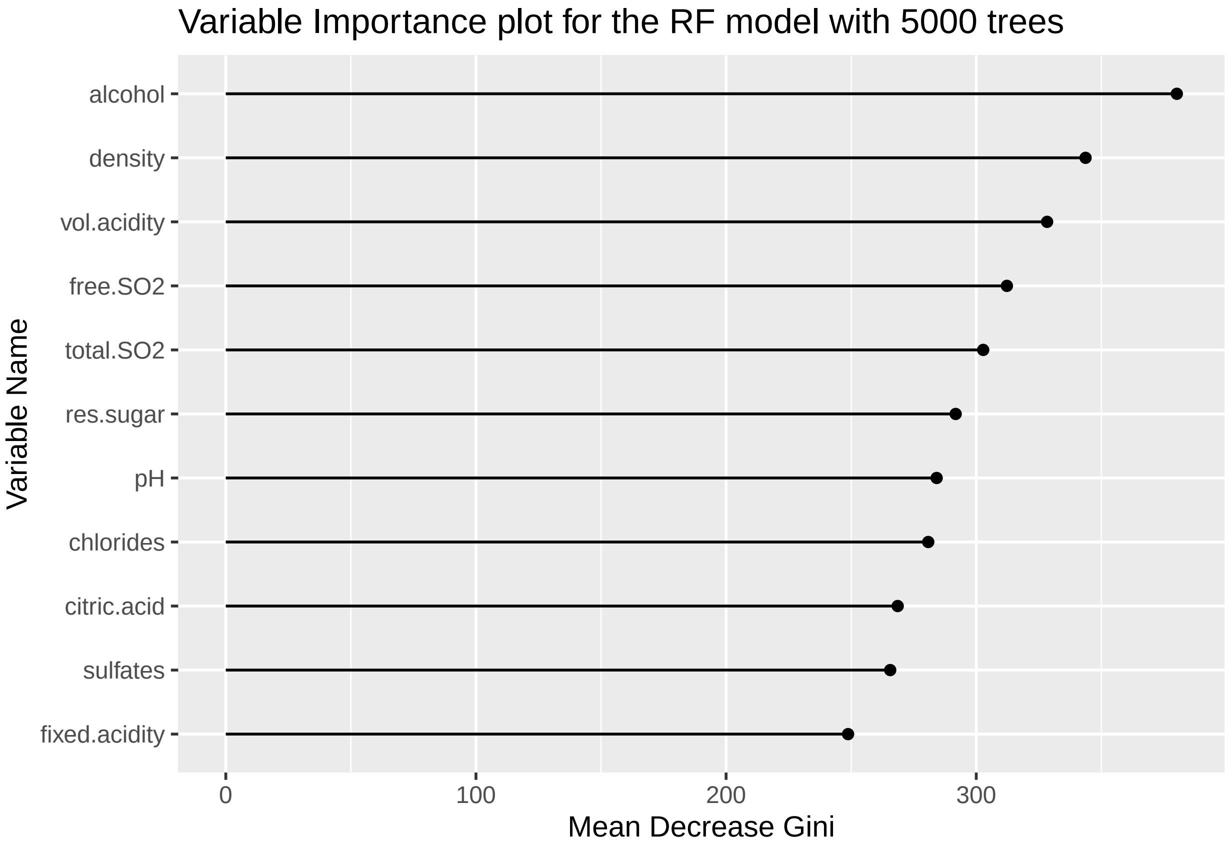
Here is the Confusion matrix for RF model:

Truth	Prediction							Class.error
	3	4	5	6	7	8	9	class.error
3	0	0	6	14	0	0	0	1.00
4	0	41	76	45	1	0	0	0.75
5	0	8	1049	391	9	0	0	0.28
6	0	4	251	1833	108	2	0	0.17
7	0	0	13	342	520	5	0	0.41
8	0	0	1	54	39	81	0	0.54
9	0	0	0	3	2	0	0	1.00

According to the confusion matrix, wines with `quality` of 3 and 9 are all predicted wrong, since their classification error is 100%. This may be due to the fact that there are not as many observations for groups

3, 9 as we have for wines with qualities of 5, 6, 7 and 8. Thus, I can expect a low prevalence of observations with these qualities in each bag and a split in the tree resulting in a prediction of a quality of 3 or 9 is unlikely.

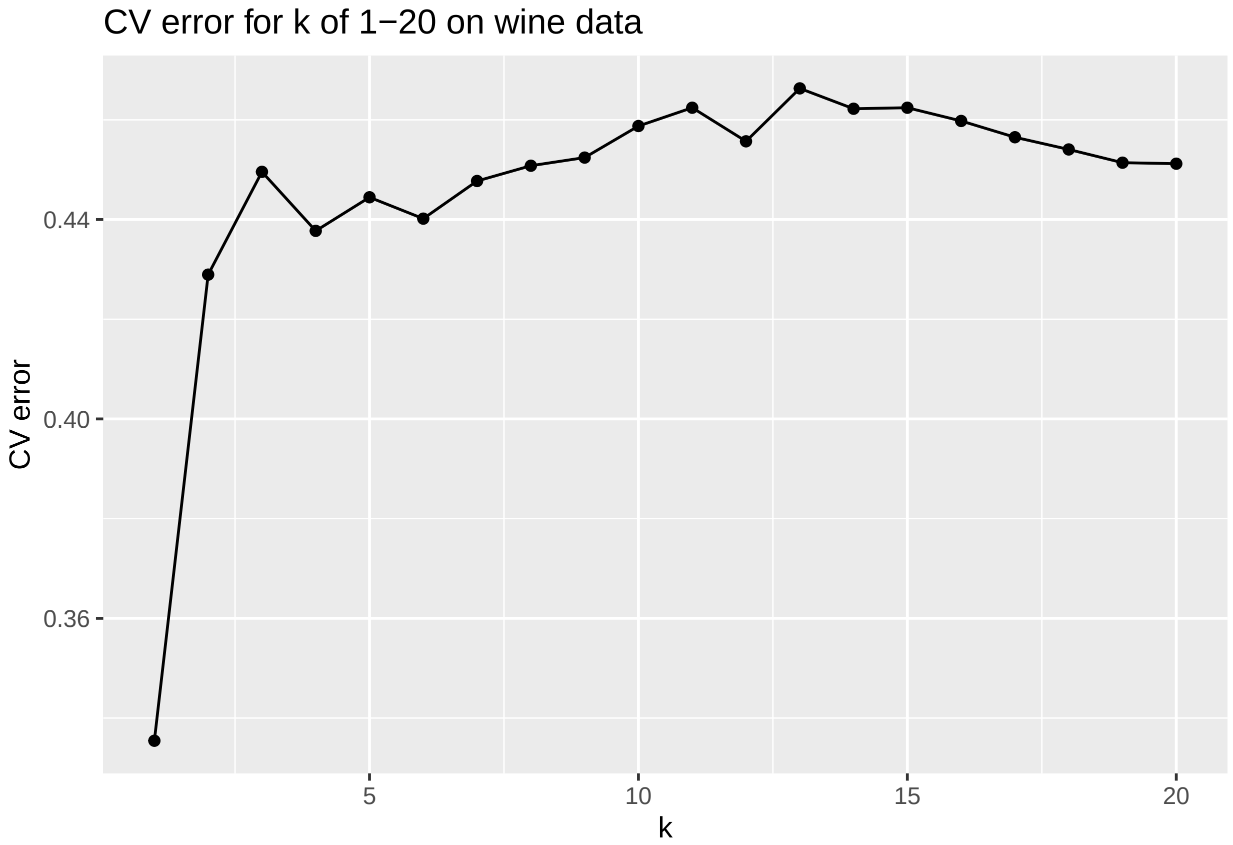
To find out what are the factors that influence the quality of wine the most, I explore the importance of each variable in the full 5000 trees random forest model.



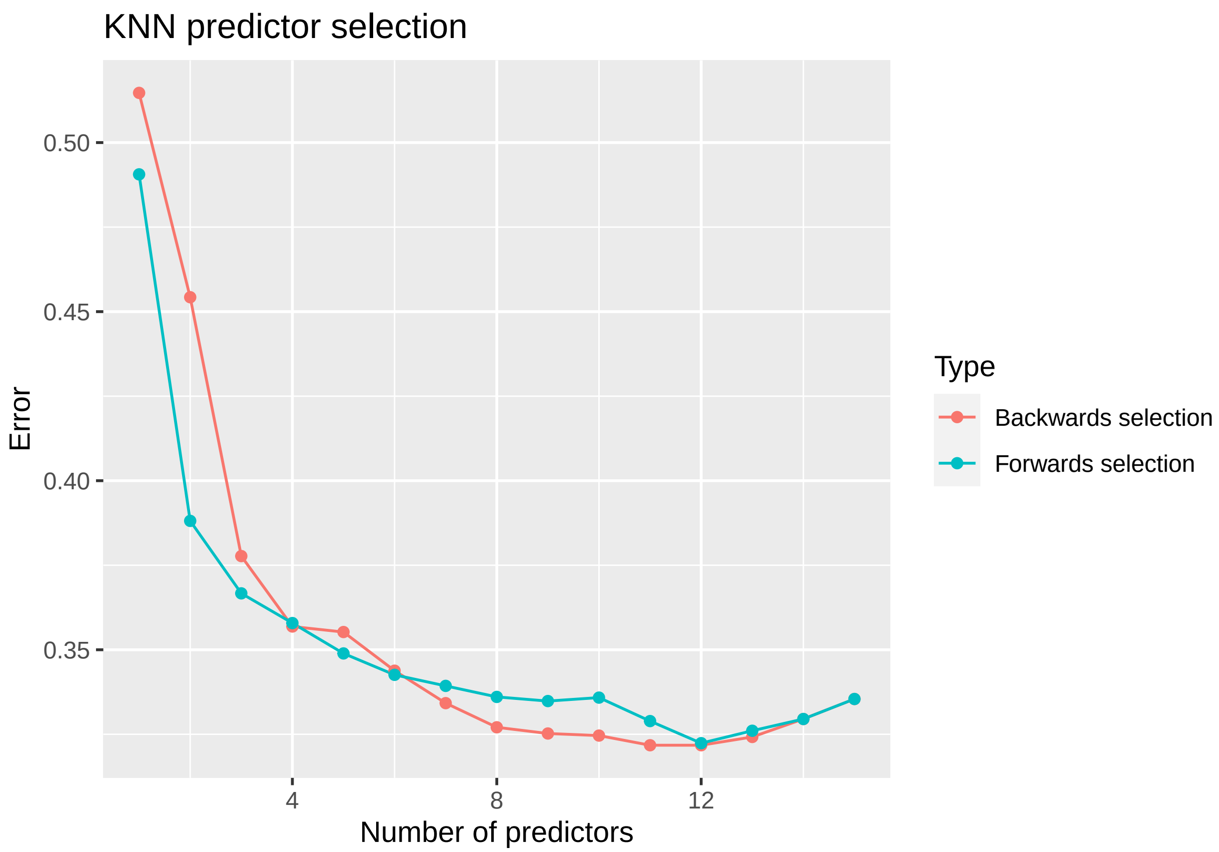
The results from the Variable Importance plot indicate that across all of the trees considered in the random forest, three most important predictors for minimizing the Gini index are `alcohol`, `density` and `vol.acidity`.

K-Nearest-Neighbor

The following chart shows the LOO-CV error of a K-Nearest Neighbors model on all 15 predictors (including the composite ones) which have been normalized. Of note is that `k=1` is by far and away the strongest `k` value. This makes sense as I expect to experience the "curse of dimensionality" in this case and the 2nd and 3rd nearest neighbors for each one may be very far away from the test value. It should be expected that `k=1` will be superior for high dimensional models.



However, it is unlikely that the optimal KNN model is that with all the predictors, especially when one considers the issues with KNN in high dimensional predictor space. Searching all 2^{15} possible models is computationally unfeasible. Thus, backwards and forwards greedy selection are used to generate a new model.



The above graph shows the CV error by number of predictors for backwards and forwards selection on a 1-nearest-neighbor model. Higher `k` values were tested but were too computationally intensive and less error. With a test of `kmax=25`, only the models with 1-3 predictors benefited from more than 1 nearest neighbor and their CV error was still far too high. It appears that backwards selection produced a model with lower prediction risk. Here are the CV-errors for each selection method:

Method	CV	k
Backwards selection	0.322	1
Forwards selection	0.322	1

Here is the Confusion matrix for the best KNN:

Truth	Prediction						
	3	4	5	6	7	8	9
3	0	3	8	9	0	0	0
4	2	57	58	39	7	0	0
5	3	43	1015	343	45	7	1
6	3	29	317	1591	227	30	1
7	0	5	40	233	560	41	1
8	0	0	10	30	36	99	0
9	0	0	1	0	3	1	0

The best model by prediction risk from cross validation is the model from both backwards and forwards selection with `k=1` and 12 predictors, all predictors except for `chlorides`, `density`, `acidity.ratio` and free.sd.ratio. The CV score for this model was 32.18% which corresponds with an estimated classification accuracy of 67.82% which is rather inaccurate. The confusion matrix, shown above, indicates that not a single wine of true quality 3 or 9 was predicted accurately. This makes sense as the data set is heavily unbalanced and if no two of the few wines in those classes are close together, we won't see a single accurate prediction. This phenomenon illustrates the limitations of using KNN on unbalanced data.

Neural Network

To constrain the scope of the project to a reasonable size and to reduce computation time, only single hidden layer neural networks were explored. To estimate prediction risk, 5-fold CV was chosen. The value of 5 was chosen as a balance between the larger variance of using less folds and the larger computational requirements of using more folds. After varying the hyperparameters, a model with a hidden layer of size 100 was chosen for the balance between computational intensity and prediction risk. However, this model under performed, with an accuracy of only 59.7%. Thus, it appears that single layer neural networks are unsuitable for this challenge.

Here is the Confusion matrix for Neural Network:

Truth	Prediction						
	3	4	5	6	7	8	9
3	0	5	8	4	3	0	0
4	1	38	74	41	9	0	0
5	7	58	952	365	63	12	0
6	9	50	412	1375	284	65	3
7	5	8	63	256	484	60	4
8	1	4	7	35	49	77	2
9	0	0	0	3	1	1	0

The confusion matrix of the neural network, shown above, displays a similar phenomenon as seen in the random forest model and KNN. The prediction accuracy of wines of quality 3 and 9 is 0. The small number of observations with these qualities used when training the model means that unless the test observation(s) happen to be extremely close to the training observations for a given rare quality, the model won't be able to predict that quality with any accuracy.

Conclusion

Final model

Out of all models explored, random forest was the best, measured by OOB error (and CV error for the others) with a prediction risk of around 28.1%, corresponding with an accuracy of around 71.9%. There are likely many reasons that the random forest excels in this case including its ability to handle correlated predictors, resulting in a lower variance while maintaining most of the low bias characteristics of trees. Further attempts to create a better model with this data could benefit from an exploration of an error function based on category distance, neural nets with more layers and ensemble models.

Are all the 4898 data points useful? (ie. missing or logically unsound)

All observations appeared to be sound. However, there were some outliers that had to be taken into account when considering how to standardize data. Furthermore, the data was rather imbalanced, leading to low classification accuracy for wines of quality 3 (extremely low) or 9 (extremely high) in KNN, RF and NN. It appears that this data was only sufficient for predicting wines of average quality (4-8).

Which are the factors that influence the quality of wine? (all or only some)

Backwards predictor selection excluded `chlorides`, `density`, `acidity.ratio` and `free.sd.ratio` from the best KNN model. However, all predictors appear to have significant importance based on the random forest variable importance plot. Thus, it is possible that the effect of the excluded predictors (the last two being composite predictors) is negligible but we cannot make any conclusions. It is likely that most, if not all, of the predictors included in the KNN model have some affect on the quality of the wine as they all increased the cross validation score.

Are there any interactions between the factors?

Yes. As discussed in the data analysis, there is a strong positive correlation between `res.sugar` and `density` and a strong negative correlation between `alcohol` and `density` as well as some other weaker relationships. As random forest decorrelates predictors by only using a subset of them for each tree, it is likely that it was so successful in part due to its ability to deal with the interactions between factors.

Are there any confounding factors? (Factors that are not considered but may influence wine quality significantly)

Some of the generated features were marginally useful when calculating KNN. However, adding them to the random forest, the highest performing model, did not decrease OOB error. Thus, I did not not discover any other factors of note. However, an interesting extension of this project would be to attempt PCA and kPCA on this dataset to try and generate principal components of some meaning.

References

Cortez, Paulo, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. 2009. "Modeling Wine Preferences by Data Mining from Physicochemical Properties." *Decision Support Systems* 47 (4): 547–53. https://doi.org/https://doi.org/10.1016/j.dss.2009.05.016.