Wine Quality:

A Machine Learning Approach to Model the Colour of Wine and its Quality Based on Physicochemical Tests

Machine Learning for Data Science

Durham University 2023-2024

12/02/2024

Student # 001130188

Introduction

The problem that this project investigates is how machine learning techniques can be used to first predict the colour of wine based on its physiochemical properties and then using that predicted colour value, combined with its original physiochemical properties, to estimate the quality score that wine would receive. Imagine you are presented a vintage case of Portuguese Vinho Verde wine in a dark bottle. All you can do is measure the physiochemical properties of this wine to try and guess what colour it is and what it might score to a professional wine taste. How might you go about achieving this? By harnessing categorization and decision trees, this type of machine learning algorithm would allow vineyards to reverse engineer our results by understanding the levels and tolerances of key physiochemical levels that lead to high quality wine. To define our terms, the quality of wine would be compared to the quality score given from a wine expert on a scale from 1-10. With a better understanding of what physiochemical tolerances to produce a wine with, wine makers can fine tune their processes to produce higher quality wine.

To accomplish this, we can look to two datasets retrieved from the UC Irvine Machine Learning Repository(Cortez). These two datasets are structured identically to each other but are separated by the colour of the wine; one being a red Portuguese vinho wine and the other being the counterpart white Portugeuese vinho wine. Each data set lays out the physiochemical properties and quality score for the wine types. The dataset included two separate CSV files; one for red wine and one for white wine. Each data set contains the following twelve variables as their columns:

1. fixed.acidity
2. volatile.acidity
3. citric.acid
4. residual.sugar
5. chlorides
6. free.sulfur.dioxide
7. total.sulfur.dioxide
8. density
9. pH
10. sulphates
11. alcohol
12. quality

The first eleven of these variables represent the physiochemical properties of the wine and are the predictor variables. The twelfth variable, quality, is our predictor variable. The white wine data set has 4898 observations while the red wine dataset has 1599 observations.

For the modelling approach that would best be used, I'll first be using classification trees to determine if, based on the physiochemical properties, it is a red or a white wine. Once I have these predicted colour values, I will stack this models predicted values into new data set which will use regression trees to determine the most important physiochemical properties that lead to high quality wine, while also showing the specific tolerance levels and decisions that will lead to the highest quality wine. Finally, to improve the performance of this model, I will utilize random forests to help improve the accuracy and reduce overfitting. The reason a CART approach works with this data set is because quality, our target variable, is bucketed into groups of numbers, rather than a countious

vector of decimal numbers that could allow linear relationships to exist. An visualization of this can be seen in Figure 1 where alcohol is plotted against quality.

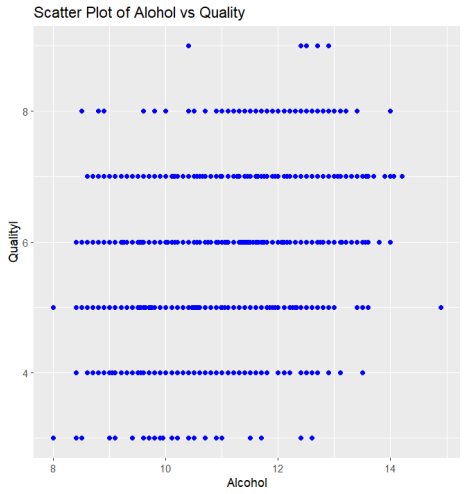

Figure 1

Each observation's alcoholic percentage falls into on of a quality category, rather than a vector made up of decimal numbers like 3.7 or 5.2. Furthermore, the decision to continue with regression trees makes sense with the projects aim to provide vineyards a strategy to formulate their wine based off a step-by-step process based off the key tolerance levels that lead to a high quality wine. Finally, the choice to choose random forests as our ensemble method instead of methods like bagging, helps give this model more confidence that it's feature selection is stronger through random forest generation.

Data Cleaning and Exploratory Data Analysis

Before building these models, the data will be inspected and cleaned. It should be noted that the source data set used ';' as the delimiter type, so in order to load these files properly, I had to specify this when reading the CSVs. After doing so, I confirmed that there are no null values for any of the variables. On top of this, all observed values follow their scheme type of a number.

Looking further into key insights from the dataset, there appears to be very low correlation between the variables(Figure 2).

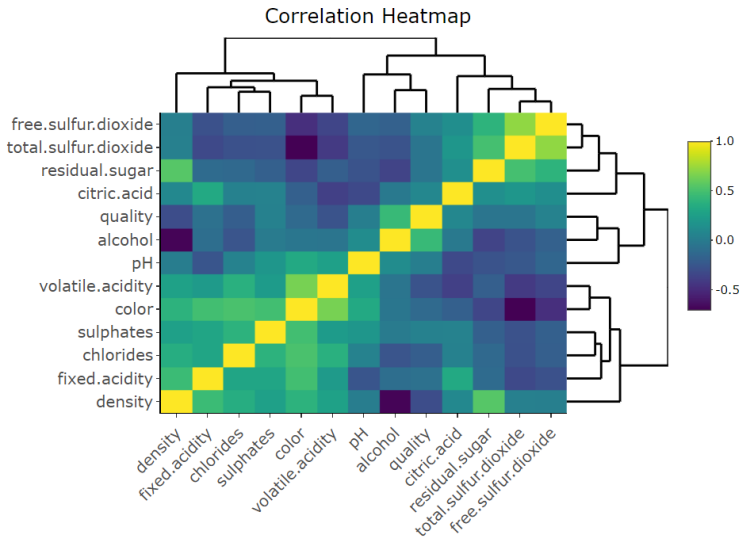

Figure 2

This is good news as low correlation among predictor variables can reduce multicollinearity, which helps improve prediction accuracy and reduce model instability.

It should also be noted that all values for these variables are continuous and numeric. The mean value for the white wine quality is 5.878 which is slightly higher than the red wine's median value of 5.636. Since part of this project's strategy is to predict the colour of wine based on the physiochemical properties, I merged the two datasets together. To know which colour the wine actually is, I added a colour column to both the red and white wine dataset with a value for 1 if it is red and a value of 0 if it is white before merging the datasets. After analysing the merged quality values between the two data sets, the mean minimum value of 3, a mean of 5.818, a maximum value of 9. Looking at the distribution of the quality score in the merged data set, it is very evenly distributed and concentrated around the mean of 5.818 (Figure 3).

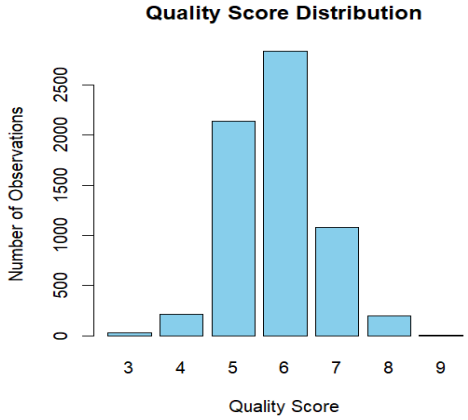

Figure 3

However, the low number of observations with a quality score of 3,4, 8, and 9 may present issues of overfitting.

Quality:	"3"	"4"	"5"	"6"	"7"	"8"	"9"
Count:	"30"	"216"	"2138"	"2836"	"1079"	"193"	"5"

For this reason, I chose to remove these outliers, resulting in a new quality range that spans from 5 to 7. The reason that these outliers needed to be removed, apart from preventing against overfitting, is when we'll use random forests to improve our regression trees, the factors from the testing and training data must match, and if one of these sub datasets contains observations observed qualities scores while the other one doesn't, it may present issues. It is for these reasons that condensing the quality score into three categories: 5, 6, and 7, makes sense to continue with CART.

Building the Model: CART

First building my classification model, I initiate it by using rpart targeting the eleven predictor variables with my all_wine dataset, the combination of both red and white wine. I then find the best split of the model by first copying the all_wine dataset to a new dataset, BSplit, with a pred_split column containing values of red or white. Using this new BSplit dataset, I build a classification tree using volatile.acidity, total.sulfur.dioxide, chlorides. The reason I chose these variables is due to their polarising correlation with colour. When analysing the correlation matrix among for the colour column among predictor variables in the original dataset, I wanted variables that had the strongest corresponding correlation with colour. These polorazing variables will help our categorization tree predict the colour with more ease since the values of these variables are highly correlated with a corresponding colour. Their correlation values are as follows:

volatile.acidity(0.65303559), total.sulfur.dioxide(-0.70035716), chlorides(0.51267825).

After building this classification model using rpart and specifying it's method as "class", we can prune it by first calculating the index of the minimum cross-validated error in the complexity table and storing it as 'opt'. We can then assign cp as the complexity parameter corresponding to the minimum cross-validated error. Once we have all of this, we can store the complexity parameter value associated with the minimum cross-validated error as cat_cp and prune the model with this value. The result is an impressive categorization model. This graph(Figure 4) illustrates a strong predictive performance. The results indicate that when it comes to predicting colour from physiochemical attributes, the first best step is to see if the total.sulfure.dioxide content is above or below 67.5.

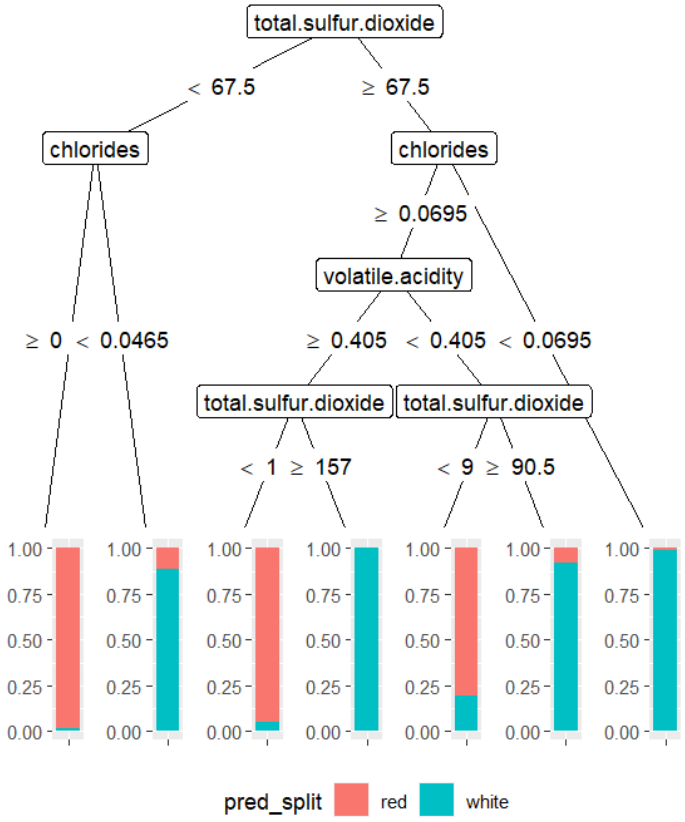

Figure 4

After creating this model, I'm curious to see how this new predicted colour value impacts the regressive tree model's ability to predict the wines quality class. To see how this plays out, we can first create a new dataset with the predicted colour values from our classification tree. We can then create two different regression tree models: the first one without the predicted colour variables and the second with the predicted colour variables included. The structure for each of these models will essentially be the same, the only difference being the datasets that they are pulling from. The first model will pull from the original merged dataset without the prediction model while the second uses the newly created dataset with the predicted colour values.

To build these regression tree models, we'll first use rpart and have each model pull the data from their corresponding data set. We can then prune the model using the same method outlined in the classification modelling to get our regression tree(Figure 5). After doing so, we'll cross validate this model using k-fold cross-validation where k = 10.

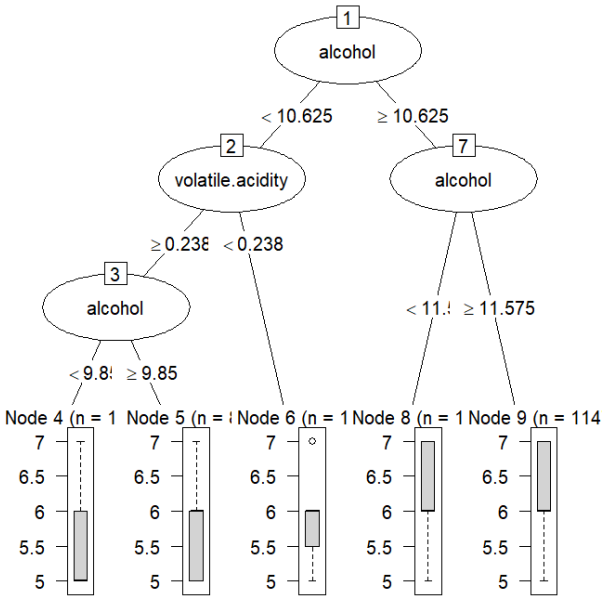

Figure 5

The results from the model with the predicted colour variables and without them were essentially the same, and they produced the same regression tree. Both models have the same optimal value for cp (0.02098693) and similar performance metrics, with the first model having a slightly lower RMSE(0.6159865 vs. 0.6182171) and a higher Rsquared value(0.2432178 vs. 0.2380588). So, stacking the predicted colour results produced a negligible decrease in the regression tree's ability to predict the quality based on physiochemical properties.

Building the Model: Random Forests

To create an ensemble model on top of our existing regression tree model, I first decided to not include the predicted colour value since it showed a negligible decrease in the regression tree's performance. I then primed our testing and training data by setting the index to choose a sample size of 50 observations each time a new subset is drawn. Afterwards, I created the testing and training data using that index size and built the random forest using volatile.acidity, total.sulfur.dioxide, chlorides, and alcohol as the predicting variables. I tinkered with the predictor variables used for building these random forests and found that using the original three variables that correlated highest with colour as well as alcohol due to it's key role in the regression model's prediction. Using these selected variables, I build my random forest using the training data and predict the values using my testing data.

Model Comparison

Comparing the initial regression tree models based on it's inclusion of predicted colour value, the model that excluded colour type with 11(Figure 6), exhibited suboptimal performance as evidenced by relatively high root mean error squared(Ranges from 0.6159865 to 0.6782277) and a low R-squared(Ranges from 0.1624267 to 0.2432178) value.

```
> rpartFit
CART

6053 samples
  11 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 5448, 5449, 5447, 5447, 5448, 5448, ...
Resampling results across tuning parameters:

  cp          RMSE       Rsquared   MAE
  0.02098693  0.6202577  0.2321841  0.5222963
  0.04902017  0.6329477  0.2009939  0.5448663
  0.18097188  0.6728769  0.1589820  0.5735288

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was cp = 0.02098693.
```

Figure 6

Comparing it to the second model,pred_rpartFit, that included the predicted colour variable type(Figure 7), both models show a limited ability to accurately predict the response variable without any further ensemble model.

```
> pred_rpartFit
CART

6053 samples
  11 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 5448, 5449, 5448, 5447, 5448, 54
Resampling results across tuning parameters:

  cp          RMSE       Rsquared   MAE
  0.02098693  0.6177718  0.2389327  0.5176901
  0.04902017  0.6376842  0.1893554  0.5521062
  0.18097188  0.6780369  0.1612334  0.5741414

RMSE was used to select the optimal model using the
 smallest value.
The final value used for the model was cp = 0.02098693.
```

Figure 7

Although the second model's performance was very similar to the first models, the slight decrease in key analytics like RMSE and R-squared led to the predicted colour value not being included in any further ensemble models.

The random forest model(Figure 8), with its addition of alcohol as a predictor variable, demonstrated a pretty large improvement in predictive performance compared to the individual rpartFit model. The confusion matrix showed improved accuracy, sensitivity, and specificity across the quality values. Specifically, the random forests helped achieve an accuracy of 76% which is a large jump from the initial model.

```
> confusionMatrix(pred, test_wine$quality)
Confusion Matrix and Statistics

          Reference
Prediction  5  6  7
         5 22  3  0
         6  5 15  4
         7  0  0  1

Overall Statistics

               Accuracy : 0.76
                 95% CI : (0.6183, 0.8694)
    No Information Rate : 0.54
    P-Value [Acc > NIR] : 0.001124

                  Kappa : 0.5677

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 5 Class: 6 Class: 7
Sensitivity            0.8148   0.8333   0.2000
Specificity            0.8696   0.7188   1.0000
Pos Pred Value         0.8800   0.6250   1.0000
Neg Pred Value         0.8000   0.8846   0.9184
Prevalence             0.5400   0.3600   0.1000
Detection Rate         0.4400   0.3000   0.0200
Detection Prevalence   0.5000   0.4800   0.0200
Balanced Accuracy      0.8422   0.7760   0.6000
```

Figure 8

This jump in improvement has large implications for those looking to reverse-engineer the results of this model when producing wine. Without random forests, using the results from the initial model would prove of very little benefit with such a low R-squared. However, with the large improvement of accuracy, insights from this model are much more feasible.

Results and Conclusion

The results from my initial classification model shows a strong ability to predict the colour of a wine based on the physiochemical properties, specifically when focusing on the levels of volatile.acidity, total.sulfur.dioxide, and chlorides. Stacking those results into a regression tree, there was no notable improvements, suggesting that the predicted wine colour does not influence the predicted quality of the wine compared to a model that just uses the physiochemical properties. Furthermore, when using random forests as my ensemble method and incorporating alcohol as a predictor variable demonstrate a strong accuracy in predicting the quality of wine based of it's physiochemical properties. This result provides winemakers tangible variables and tolerances to consider when trying to improve the quality of their wine to increase it's appeal.

As for the limitations of the data, a key pain point was how the quality values were categorized into integer values rather than decimal numbers. This forces the dataset away from any linear regression models that could have provided further key correlation and predictive insights. It

also may have forced more datasets to be removed which can hurt the model's performance when presented with physiochemical properties that actually make up a very high or low quality wine. For example, if the data were set up with decimal numbers, wine given a quality score around 4.4 may have been included in our data set, but since it may have been rounded down to 4 in the critics' evaluation, it isn't considered which reduces the size of our dataset.

As for the limitations of the approaches used, CART models can be prone to overfitting, which can be seen in the low R-squared values from my initial regression tree models, leading to poor prediction with new data. Random Forests on the otherhand, while designed to mitigate overfitting, might still suffer from bias introduced by the underlying decision trees.

One suggestion moving forward to provide a more robust predictive model would be for the quality variable to be assessed as decimal numbers so that any linear relationships or insights can be explored.

Citations

1. Cortez, Paulo, Cerdeira,A., Almeida,F., Matos,T., and Reis,J.. (2009). Wine Quality. UCI Machine Learning Repository. https://doi.org/10.24432/C56S3T.