Wine Quality Capstone Project

Manuel_5600 (edx username)

18th of March 2019

Contents

Introduction/Overview/Summary **2**
- Data Set Structure - General Overview . . . 2
- Distribution and Correlation - General Overview . . . 3
- Regression/Classification Trees and Random Forests - General Overview . . . 4

Data Set: White Wine - Rating on a Scale from 0 to 10? **5**
- Methods/Analysis/Results . . . 6
- Confusion Matrix . . . 7
- Error Investigation . . . 8

Data Set: White Wine - "Good", "Average" or "Bad"? **9**
- Distribution and Correlation . . . 9
- Methods/Analysis/Results . . . 10
- Confusion Matrix . . . 11
- Error Investigation . . . 12

Data Set: White Wine - "Good" or "Bad"? **13**
- Distribution and Correlation . . . 13
- Methods/Analysis/Results . . . 14
- Confusion Matrix . . . 15
- Error Investigation . . . 16

Data Set: White Wine - "Good" or "Bad"? - Attempt to Balance **17**
- Distribution and Correlation . . . 17
- Methods/Analysis/Results . . . 18
- Confusion Matrix . . . 19
- Error Investigation . . . 20

Data Set: Complete Data Set - White or Red Wine? **21**
- Data Set Structure . . . 21
- Distribution and Correlation . . . 22
- Methods/Analysis/Results . . . 23
- Confusion Matrix . . . 24
- Error Investigation . . . 25

Conclusion **26**
- Random Forest 1 / White Wine - Rating on a Scale from 0 to 10? . . . 26
- Random Forest 2 / White Wine - "Good", "Average" or "Bad"? . . . 26
- Random Forest 3 / White Wine - "Good" or "Bad"? . . . 26
- Random Forest 4 / White Wine - Possible Limitations . . . 26
- Random Forest 5 / Complete Data Set - White or Red Wine? . . . 26

Introduction/Overview/Summary

The two data sets are available on the UCI machine learning repository (Source: https://archive.ics.uci.edu/ml/datasets/wine+quality) and related to red and white wine. Each row contains a quality score ranging from 0 to 10 and eleven physicochemical properties of the corresponding wine sample. Firstly, the goal was to predict a wine's quality based on its wine properties and secondly to predict if these properties belong to a red or a white wine. In a nutshell, the train function with the method Rborist and the tuning parameters predFixed and minNode was used. Furthermore, the wine quality ratings were grouped and the data was balanced in multiple attempts to achieve higher accuracy, sensitivity and specificity. The R code is excluded from this report in order to assure better readability. However, the R and Rmd files are available to the reader.

Data Set Structure - General Overview

The structure of the two data sets is illustrated below. The physicochemical wine properties are fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates and alcohol. The quality score is based on sensory data (a score between 0 and 10 / increasing quality score = increasing quality).

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality
7.0	0.27	0.36	20.7	0.045	45	170	1.0010	3.00	0.45	8.8	6
6.3	0.30	0.34	1.6	0.049	14	132	0.9940	3.30	0.49	9.5	6
8.1	0.28	0.40	6.9	0.050	30	97	0.9951	3.26	0.44	10.1	6
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.40	9.9	6
7.2	0.23	0.32	8.5	0.058	47	186	0.9956	3.19	0.40	9.9	6
8.1	0.28	0.40	6.9	0.050	30	97	0.9951	3.26	0.44	10.1	6
6.2	0.32	0.16	7.0	0.045	30	136	0.9949	3.18	0.47	9.6	6

Note:
wine quality white / limited to 7 rows

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality
7.4	0.70	0.00	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
7.8	0.88	0.00	2.6	0.098	25	67	0.9968	3.20	0.68	9.8	5
7.8	0.76	0.04	2.3	0.092	15	54	0.9970	3.26	0.65	9.8	5
11.2	0.28	0.56	1.9	0.075	17	60	0.9980	3.16	0.58	9.8	6
7.4	0.70	0.00	1.9	0.076	11	34	0.9978	3.51	0.56	9.4	5
7.4	0.66	0.00	1.8	0.075	13	40	0.9978	3.51	0.56	9.4	5
7.9	0.60	0.06	1.6	0.069	15	59	0.9964	3.30	0.46	9.4	5

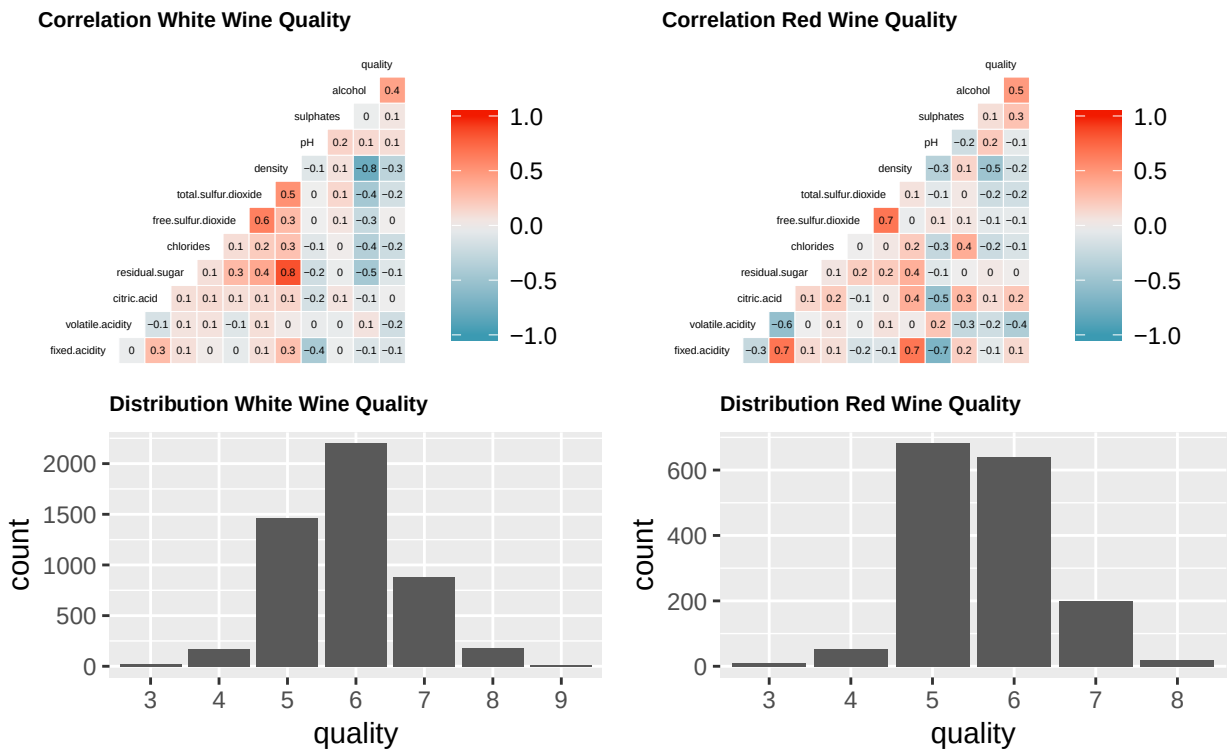
Note:
wine quality red / limited to 7 rows

Distribution and Correlation - General Overview

Distribution: The wines were mostly awarded average scores from 5 to 7. Some of the very bad and very good ratings do not exist at all (e.g. 0, 1, 2, 10) and some were only given very rarely (e.g. 9). This is not surprising due to the fact that the provider of the data set states: *"The classes are ordered and not balanced (e.g. there are much more normal wines than excellent or poor ones)"* (Source: https://archive.ics.uci.edu/ml/datasets/wine+quality).

Correlation: The correlation matrix does not state very high correlation coefficients for any of the wine properties in relation to wine quality. Alcohol seems to have the highest positive correlation of 0.4 (white) and 0.5 (red) and thus seems to improve quality in both red and white wine. Moreover, an increasing amount of volatile acidity seems to decrease the quality (especially in red wine, e.g. -0.4 in red wine). Volatile acidity is gaseous and has a smell. Sweet wines and wines made from dried grapes often contain high levels of volatile acidity. Sometimes it might even add complexity and interest but in high quantities it can be disturbing (similar to the smell of vinegar or nail polish) (Source: https://www.decanter.com/learn/volatile-acidity-va-45532/). Actually, volatile acidity is a great example to highlight the complexity of sensory evaluation (it may or may not increase quality). Additionally, the data set does show high correlation within the physicochemical wine properties (e.g. 0.8 = residual sugar and density in white wine, 0.6 and 0.7 = free sulfur dioxide and total sulfur dioxide in both wine types, 0.7 = fixed acidity and citric acid in red wine, 0.7 = fixed acidity and density in red wine). Lastly, the white wine correlation matrix seems to differ from the red wine matrix. This might become important as soon as we try to predict the type of wine (red or white). Remember that a 0 indicates no correlation, -1 a perfect negative and 1 a perfect positive correlation.

Quality Plots Correlation Distribution-1.bb



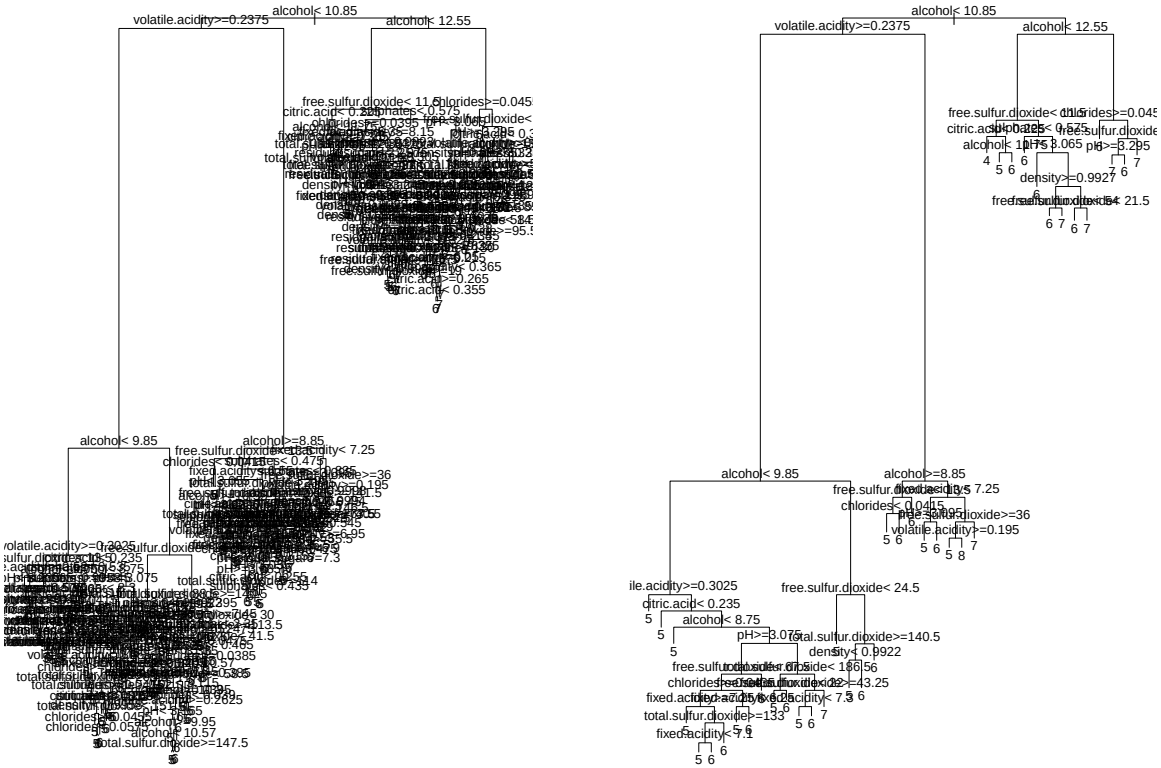
Regression/Classification Trees and Random Forests - General Overview

We all know that a forest consists of many trees. In data science a tree is a flow chart of yes/no questions. An algorithm uses data to create these trees (by partitioning the predictors) with predictions at the ends (referred to as nodes). Classification or decision trees are used for categorical outcomes and regression trees for continuous outcomes (Source: https://rafalab.github.io/dsbook/examples-of-algorithms.html#classification-and-regression-trees-cart).

The plots below are just examples to illustrate what such a tree might look like. Furthermore, the two plots proof that trees can be adjusted by tuning some of the parameters (e.g. the left tree has a complexity parameter (cp) of 0 and the right one of 0.0018, both are created with the rpart function and with minsplit of 20). Larger cp values stop the algorithm earlier which results in fewer nodes. This is due to the fact that any split which does not lower the overall lack of fit by the cp factor is not done. Minsplit is the minimum amount of observations that must exist in a node for a split to be attempted. 20 is the default value (Sources: https://rafalab.github.io/dsbook/examples-of-algorithms.html#classification-and-regression-trees-cart and https://stat.ethz.ch/R-manual/R-devel/library/rpart/html/rpart.control.html).

The random forest method is popular in machine learning. It tries to improve the prediction performance and to reduce instability by averaging decision trees. In a nutshell, a random forest is a forest of trees which has been constructed with randomness. The method generates many predictors (each using regression or classification trees) and a final prediction based on the average prediction of all the generated trees. In order to avoid identical individual trees, bootstrap is used to induce randomness (Source: https://rafalab.github.io/dsbook/examples-of-algorithms.html#classification-and-regression-trees-cart).

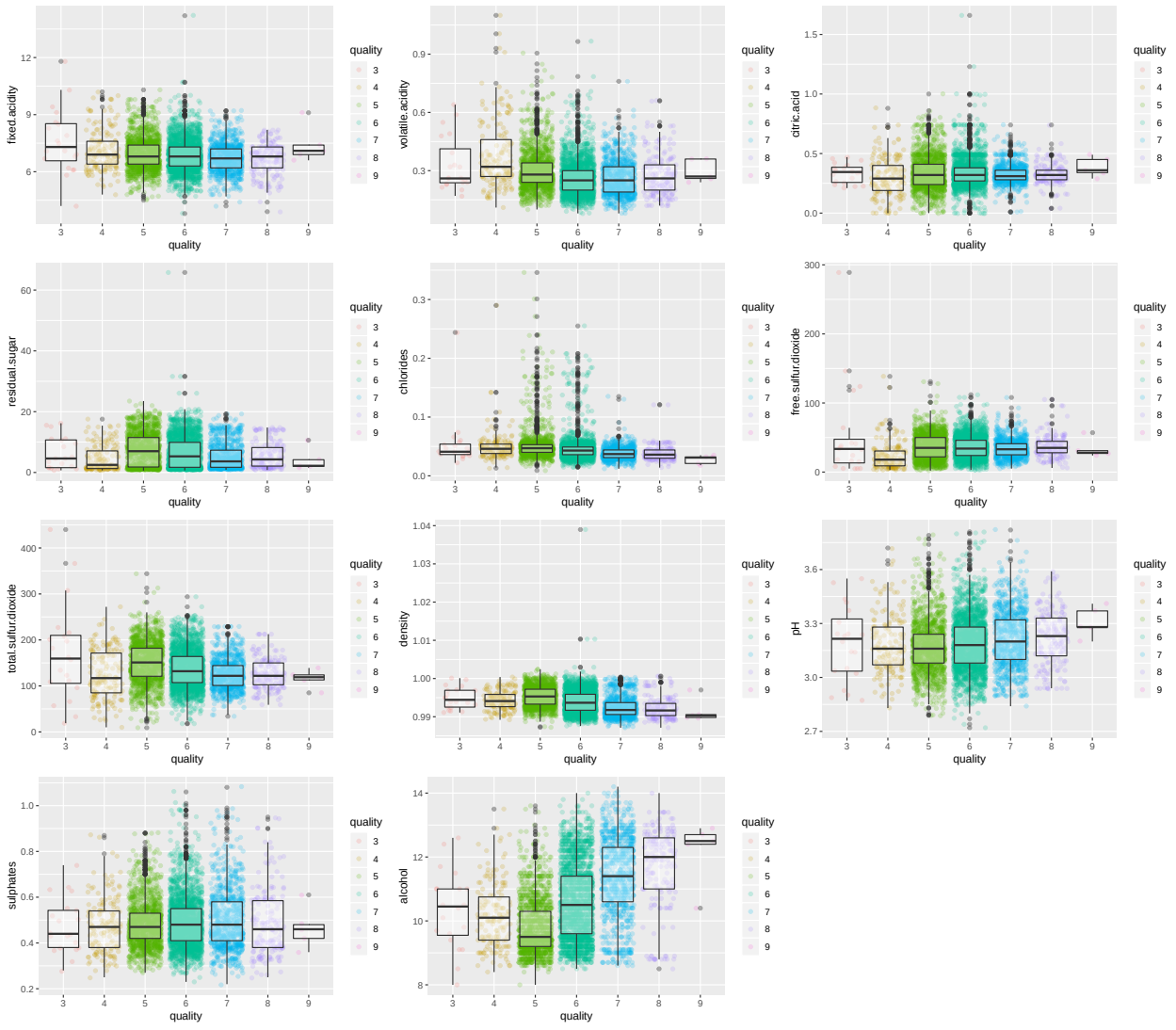
Tree-1.bb



Data Set: White Wine - Rating on a Scale from 0 to 10?

As a first step, we are going to focus on the white wine. An initial familiarization with the available data set was necessary. It is already known that the data set mainly focuses on average wines. Therefore, we might encounter a lack of data that describes the wine properties for bad and good wines. Please find below an illustration showing a combination of jitter and boxplot with quality on the x and the value of a specific wine property on the y axis. Most of the boxplots do not seem to show a clear trend (e.g. increasing from 0 to 10). However, the quality scores 5, 6, 7 that contain most of the data seem to reveal trends. Furthermore, some outliers are visible which is not surprising due to the fact that the provider of the data set mentions that *"outlier detection algorithms could be used to detect the few excellent or poor wines"* (Source: https://archive.ics.uci.edu/ml/datasets/wine+quality). We did not remove any of the outliers in the training or test data set.

Quality Boxplots-1.bb



Methods/Analysis/Results

Let's try to predict the quality ratings for white wine (rating from 0 to 10). The initial goal was to optimize overall accuracy (the overall proportion that is predicted correctly).

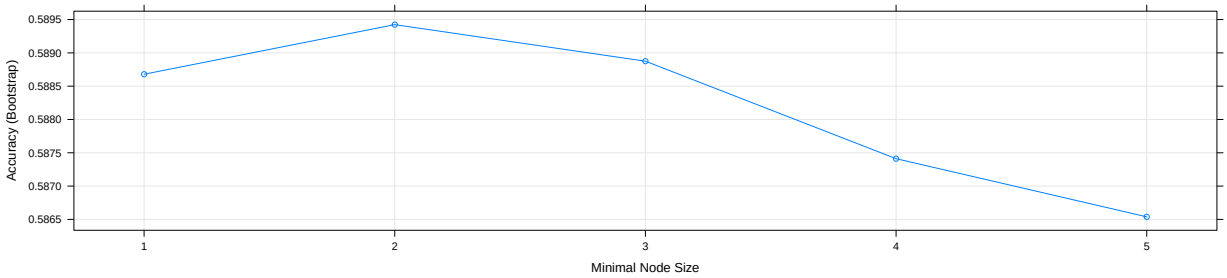
The caret train function was used and Rborist was chosen as training method for the random forest. It offers the tuning parameters predFixed and minNode. PredFixed is the actual number of predictors for a split. The default value seems to be 0. The minNode parameter is a lower limit for node sizes, which is the amount of distinct samples subsumed by each node. A value of 1 allows splitting until purity. A larger value results in a smaller tree and faster training (Source: https://cran.r-project.org/web/packages/Rborist/vignettes/rborist.html). The plot below shows the development of overall accuracy for changing values of minNode. The table best tune shows the combination of predFixed and minNode that resulted in the highest overall accuracy. Tuning of predFixed did not seem to cause major accuracy improvements but it increased training time substantially. Furthermore, we tried using knn (k nearest neighbor) with the tuning parameter k, which did not result in a major improvement as well.

Moreover, the two properties free sulfur dioxide and citric acid which show a correlation of 0 in the white wine correlation matrix were not removed (removal did actually cause a decrease in overall accuracy). Another trial for which we only used density (−0.3 correlation) and alcohol (0.4 correlation) also resulted in a decrease of overall accuracy. This is why none of the eleven properties were removed. Furthermore, the variable importance of Rborist does not show a variable that is clearly more important.

	predFixed	minNode
2	0	2

Note:
best tune

Forest Quality 1 to 10-1.bb




```
## Rborist variable importance
##
##                      Overall
## volatile.acidity     0.05935
## residual.sugar       0.05189
## free.sulfur.dioxide  0.05182
## alcohol              0.04701
## density              0.04568
## total.sulfur.dioxide 0.04541
## citric.acid          0.04340
## fixed.acidity        0.03768
## chlorides            0.03216
## pH                   0.02964
## sulphates            0.02517
```

```
## [1] 0.6332109
```

The random forest predicted the quality ratings with an overall accuracy of almost two-thirds (see exact number above). This is not too bad for our first attempt but seems to be of limited usefulness.

Confusion Matrix

Overall accuracy can be a deceptive measure. Biased data sets are split into biased training and test sets and therefore most likely result in a biased algorithm. An improvement to exclusively using overall accuracy is to additionally calculate sensitivity and specificity. Sensitivity is the ability to predict a positive outcome when the outcome is actually positive. Specificity is the ability to not predict a positive when the outcome is actually not a positive. A confusion matrix was constructed, which illustrates the predictions and the actual values. Furthermore, it calculates overall accuracy, sensitivity and specificity (Source: https://rafalab.github.io/dsbook/introduction-to-machine-learning.html#the-confusion-matrix).

	3	4	5	6	7	8	9
3	0	0	0	0	0	0	0
4	0	11	2	0	0	0	0
5	4	51	464	187	20	0	0
6	6	19	262	867	228	47	1
7	0	1	1	45	192	23	2
8	0	0	0	0	0	18	0
9	0	0	0	0	0	0	0

Note:
confusion matrix

	x
Accuracy	0.6332109

Note:
overall accuracy

	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Precision	Recall	F1	Prevalence	Detection Rate	Detection Prevalence	Balanced Accuracy
Class: 3	0.0000000	1.0000000	NaN	0.9959200	NA	0.0000000	NA	0.0040800	0.0000000	0.0000000	0.5000000
Class: 4	0.1341463	0.9991558	0.8461538	0.9708778	0.8461538	0.1341463	0.2315789	0.0334557	0.0044880	0.0053040	0.5666511
Class: 5	0.6364883	0.8478513	0.6391185	0.8463768	0.6391185	0.6364883	0.6378007	0.2974296	0.1893105	0.2962056	0.7421698
Class: 6	0.7888990	0.5835799	0.6062937	0.7727718	0.6062937	0.7888990	0.6856465	0.4483884	0.3537332	0.5834353	0.6862394
Class: 7	0.4363636	0.9641969	0.7272727	0.8866027	0.7272727	0.4363636	0.5454545	0.1795186	0.0783354	0.1077111	0.7002803
Class: 8	0.2045455	1.0000000	1.0000000	0.9712289	1.0000000	0.2045455	0.3396226	0.0359037	0.0073439	0.0073439	0.6022727
Class: 9	0.0000000	1.0000000	NaN	0.9987760	NA	0.0000000	NA	0.0012240	0.0000000	0.0000000	0.5000000

Note:
by class

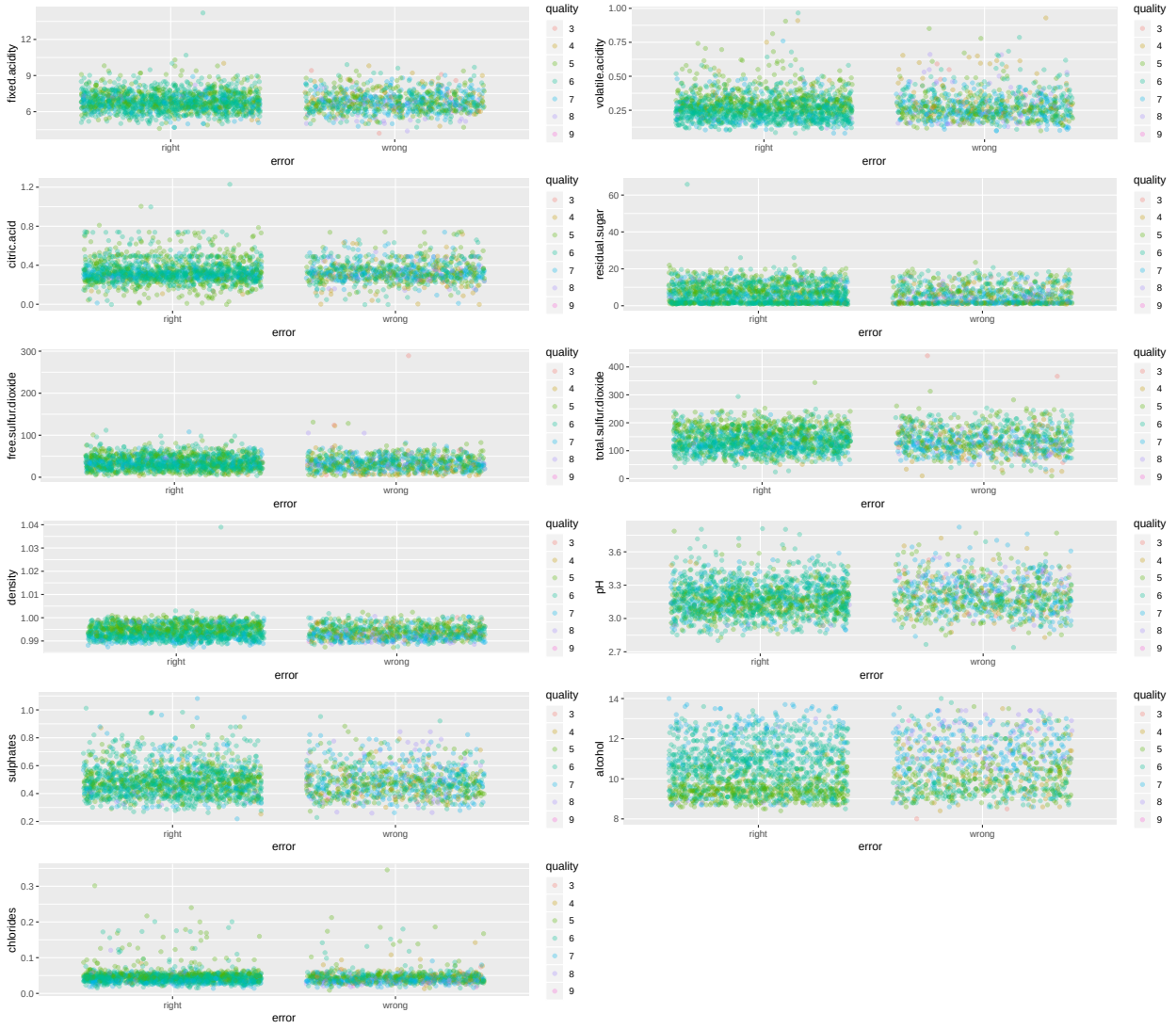
As mentioned before, the random forest predicted the eleven quality ratings with an overall accuracy of almost two-thirds (exact number mentioned above). This is not too bad but seems to be of limited usefulness. Additionally, it was mainly correct at predicting "average" wines and did a poor job at detecting "bad" and "good" wines.

Error Investigation

We tried to further investigate where the errors occurred. The plots help us to visually discover lots of wrong predictions. However, we did not find a clear pattern that would help us in the pursuit of higher accuracy (except that there are hardly any data points for some quality scores which we already discovered earlier).

```
## [1] 899
```

Plots 1 to 10-1.bb



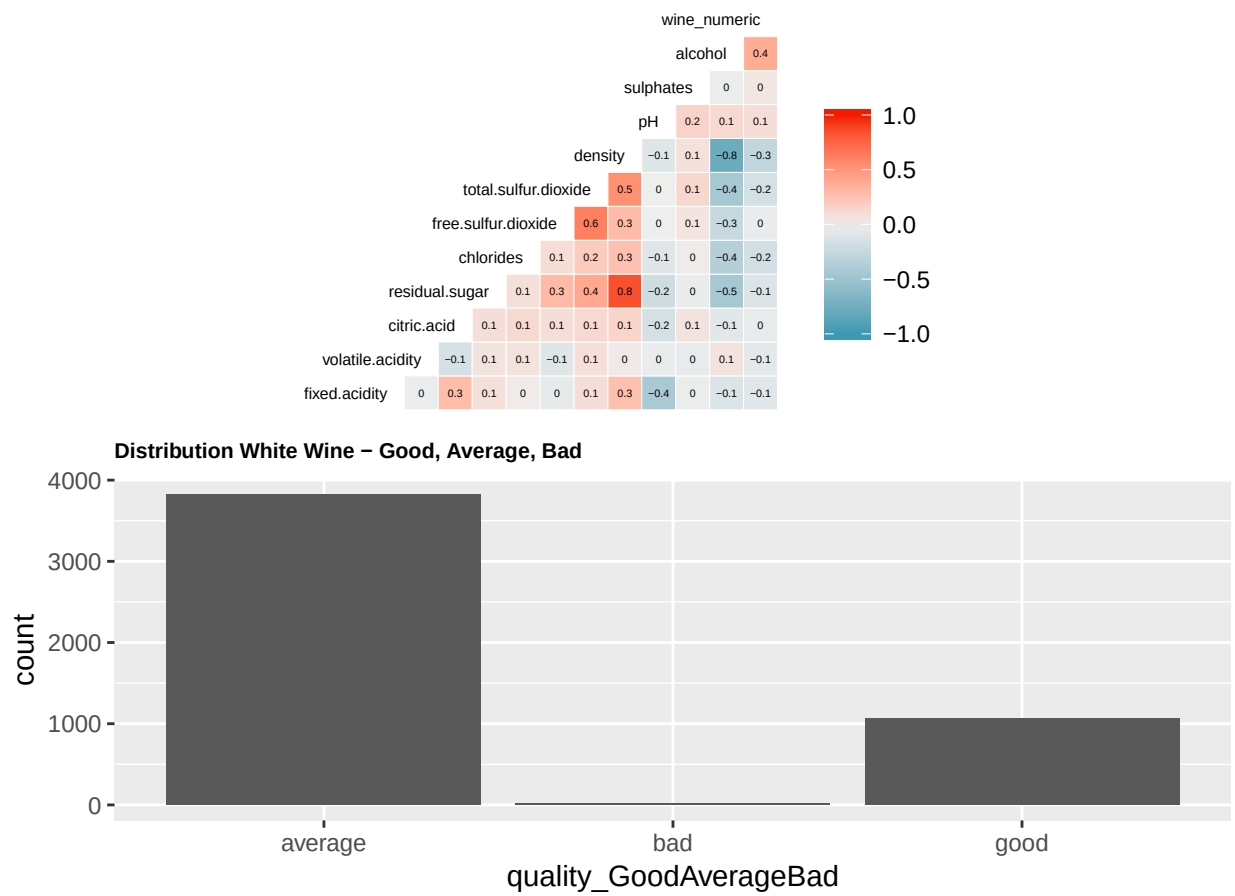
Data Set: White Wine - “Good”, “Average” or “Bad”?

Distribution and Correlation

Distribution: Our initial data exploration showed that the ratings were not equally distributed at all (e.g. some ratings were given more often than others). Let’s try to predict the quality ratings for white wine after grouping into “good” (rating 7-10), “average” (rating 4-6) and “bad” (rating 0-3). The ratings were split in a way to contain similar amounts of ratings (“bad” contains the ratings 0, 1, 2, 3 / “average” contains 4, 5, 6 / “good” contains 7, 8, 9, 10). Empty rating categories (e.g. rating 1) were not dropped due to the fact that they might suddenly exist in real life data. There was no information available on what the numerical ratings (0-10) actually mean. We only know the meaning of the extreme values (0 is equal to “very bad” and 10 to “excellent”). However, even after grouping the quality ratings into 3 groups, the group “bad” hardly exists.

Correlation: Furthermore, the grouping only caused minor changes in the correlation coefficients of the wine properties in relation to the wine quality. The correlation within the wine properties remains unchanged (same data). Remember that a 0 indicates no correlation, -1 a perfect negative and 1 a perfect positive correlation.

Good Average Bad Correlation Distribution-1.bb



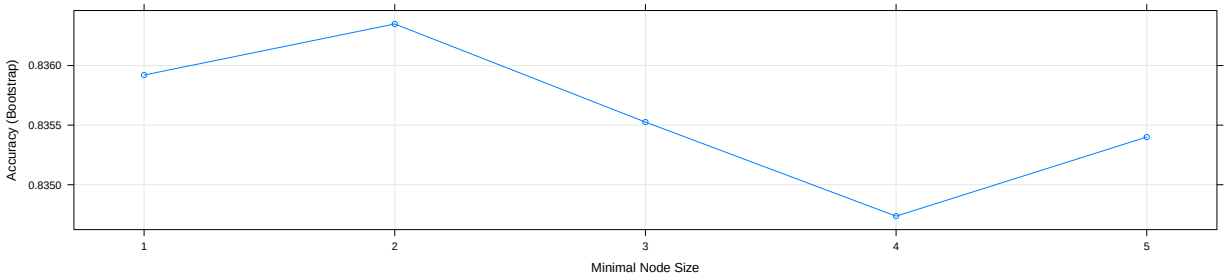
Methods/Analysis/Results

The same method was applied to the new data set (Rborist in caret train function). The table "best tune" shows the combination of predFixed and minNode that resulted in the highest overall accuracy. Tuning of predFixed did not seem to cause major accuracy improvements but it increased training time substantially. The goal was again to optimize overall accuracy, which is simply defined as the overall proportion that is predicted correctly. A change was discovered in the variable importance of Rborist. However, removal of certain variables did not cause a relevant improvement of overall accuracy. Additionally, knn (k nearest neighbor) with the tuning parameter k was tried, which did not result in a major improvement as well.

	predFixed	minNode
2	0	2

Note:
best tune

Forest Quality Good Average Bad-1.bb




```
## Rborist variable importance
##
##                       Overall
## density               0.13954
## sulphates             0.06345
## residual.sugar        0.06040
## volatile.acidity      0.05099
## pH                    0.04822
## free.sulfur.dioxide   0.03839
## chlorides             0.03814
## alcohol               0.03658
## total.sulfur.dioxide 0.03503
## fixed.acidity         0.03000
## citric.acid           0.02909
```

```
## [1] 0.8513679
```

The random forest predicted the quality ratings after grouping with an improved overall accuracy (see exact number above), which actually might indicate a very useful prediction. We do lose some information by grouping the quality scores. However, the "standard" wine drinker most likely does not care about the exact quality score on a scale from 0-10. It might be sufficient to know if he/she has to expect a "good", "average" or "bad" quality. Even though a winemaker might prefer a more detailed quality prediction, it might already be useful to know if the wine is going to rank in the "good", "average" or "bad" section of the 0-10 quality scale (based on 11 wine properties).

Confusion Matrix

We did already explain why overall accuracy can be a deceptive measure. Therefore, we constructed a confusion matrix, which illustrates the predictions and the actual values. Furthermore, it calculates overall accuracy, sensitivity and specificity.

	average	bad	good
average	1851	10	296
bad	0	0	0
good	58	0	234

Note:
confusion matrix

	x
Accuracy	0.8513679

Note:
overall accuracy

	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value	Precision	Recall	F1	Prevalence	Detection Rate	Detection Prevalence	Balanced Accuracy
Class: average	0.9696176	0.4333333	0.8581363	0.8013699	0.8581363	0.9696176	0.9104771	0.7795018	0.7558187	0.8807677	0.7014755
Class: bad	0.0000000	1.0000000	NaN	0.9959167	NA	0.0000000	NA	0.0040833	0.0000000	0.0000000	0.5000000
Class: good	0.4415094	0.9697759	0.8013699	0.8627724	0.8013699	0.4415094	0.5693431	0.2164149	0.0955492	0.1192323	0.7056427

Note:
by class

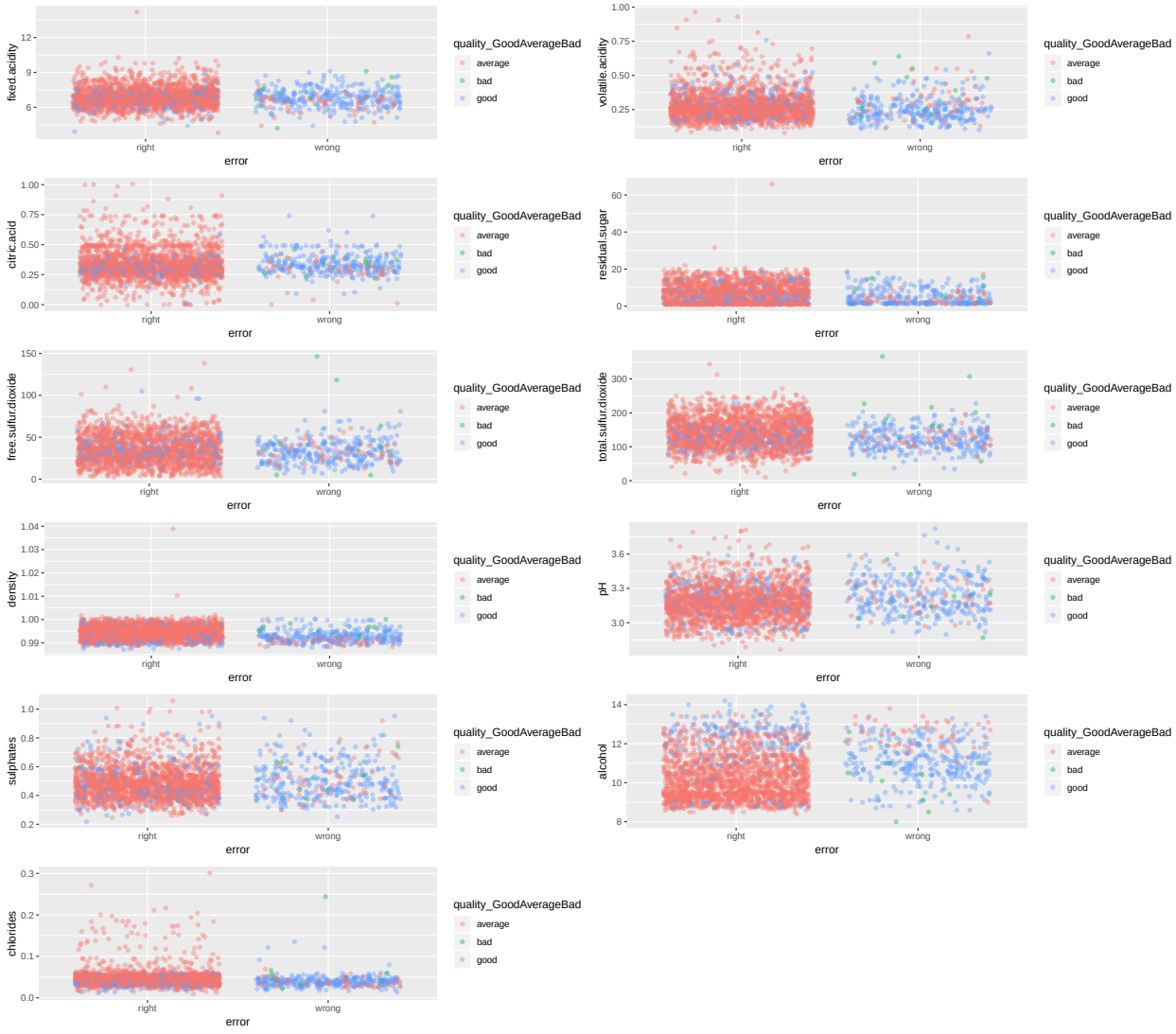
As mentioned before, the random forest predicted the quality ratings after grouping with a higher overall accuracy (see exact number above), which might indicate a very useful prediction. However, it still did a very bad job at predicting "bad" and a bad job at predicting "good" wines. Nevertheless, very good results were obtained when predicting "average" wines.

Error Investigation

We tried to further investigate where the errors occurred. In total we got fewer predictions wrong. However, many "good" wines and all the "bad" wines were rated wrong. Please remember that there were only very few "bad" wines in the data set. Our predictions for the largest group "average wine" were very accurate.

```
## [1] 364
```

Plots Good Average Bad-1.bb



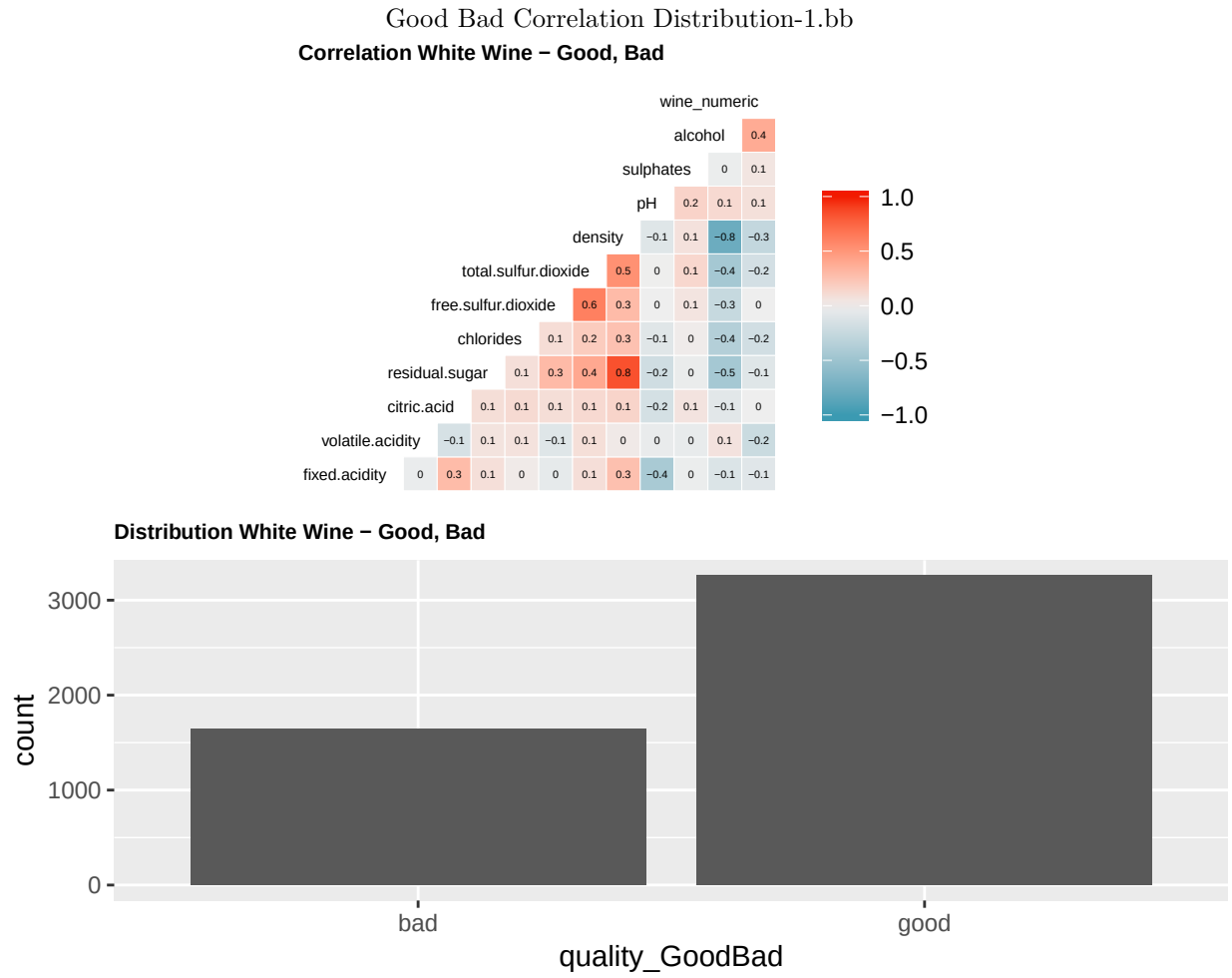
Data Set: White Wine - “Good” or “Bad”?

Distribution and Correlation

Distribution: Even the data set with three grouped ratings did hardly contain any data for “bad”. Now let’s try to predict the quality ratings for only two groups - “bad” (0-5) and “good” (6-10) wine. The new “bad” group does now contain some more data but still less than the “good” group.

Correlation: The grouping only caused minor changes in the correlation coefficients of the wine properties in relation to the wine quality. The correlation within the wine properties remains unchanged (same data). Remember that a 0 indicates no correlation, -1 a perfect negative and 1 a perfect positive correlation.

Good Bad Correlation Distribution-1.bb



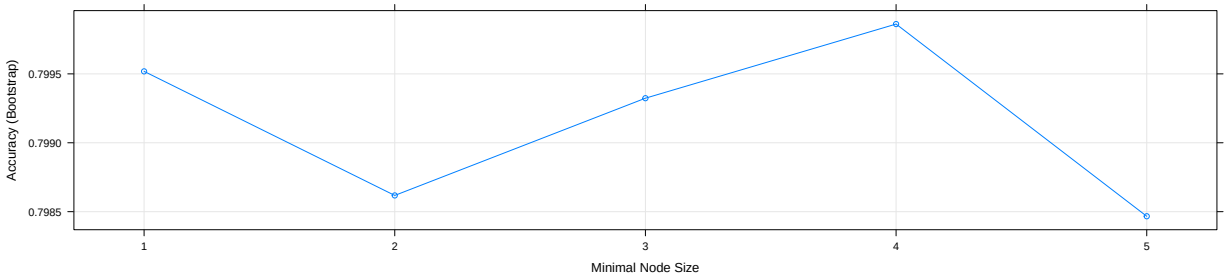
Methods/Analysis/Results

The same method was applied to the new data set (Rborist in caret train function). The table "best tune" shows the combination of predFixed and minNode that resulted in the highest overall accuracy. Tuning of predFixed did not seem to cause major accuracy improvements but it increased training time substantially. The goal was again to optimize overall accuracy, which is simply defined as the overall proportion that is predicted correctly. The variable importance of Rborist changed again. However, removing/selecting certain variables does not seem to cause major improvements in overall accuracy.

	predFixed	minNode
4	0	4

Note:
best tune

Forest Quality Good Bad-1.bb




```
## Rborist variable importance
##
##                       Overall
## alcohol               0.20343
## volatile.acidity      0.14481
## free.sulfur.dioxide   0.09782
## citric.acid           0.08613
## fixed.acidity         0.07700
## total.sulfur.dioxide 0.07676
## residual.sugar        0.06917
## density               0.05957
## pH                    0.05916
## sulphates             0.05313
## chlorides             0.05089
```

```
## [1] 0.8027766
```

The random forest predicted the quality ratings after further grouping with a slightly lower accuracy (see exact result above), which might still indicate a useful prediction. We do lose even more information by further grouping. The "standard" wine drinker most likely does not care about the exact score on a scale from 0-10. It might be sufficient to know if he/she has to expect a "good" or "bad" quality. Even though a winemaker might prefer a more detailed quality prediction, it might already be useful to know if the wine is going to rank in the "good" or "bad" section of the 0-10 quality scale (based on 11 wine properties). However, many of the bad wines would actually be nice to drink and many good wines might not be as good as promised.

Confusion Matrix

We did already explain why overall accuracy can be a misleading measure. Therefore, we constructed a confusion matrix, which illustrates the predictions and the actual values. Furthermore, it calculates overall accuracy, sensitivity and specificity.

	bad	good
bad	564	227
good	256	1402

Note:
confusion matrix

	x
Accuracy	0.8027766

Note:
overall accuracy

	x
Sensitivity	0.6878049
Specificity	0.8606507
Pos Pred Value	0.7130215
Neg Pred Value	0.8455971
Precision	0.7130215
Recall	0.6878049
F1	0.7001862
Prevalence	0.3348305
Detection Rate	0.2302981
Detection Prevalence	0.3229890
Balanced Accuracy	0.7742278

Note:
by class

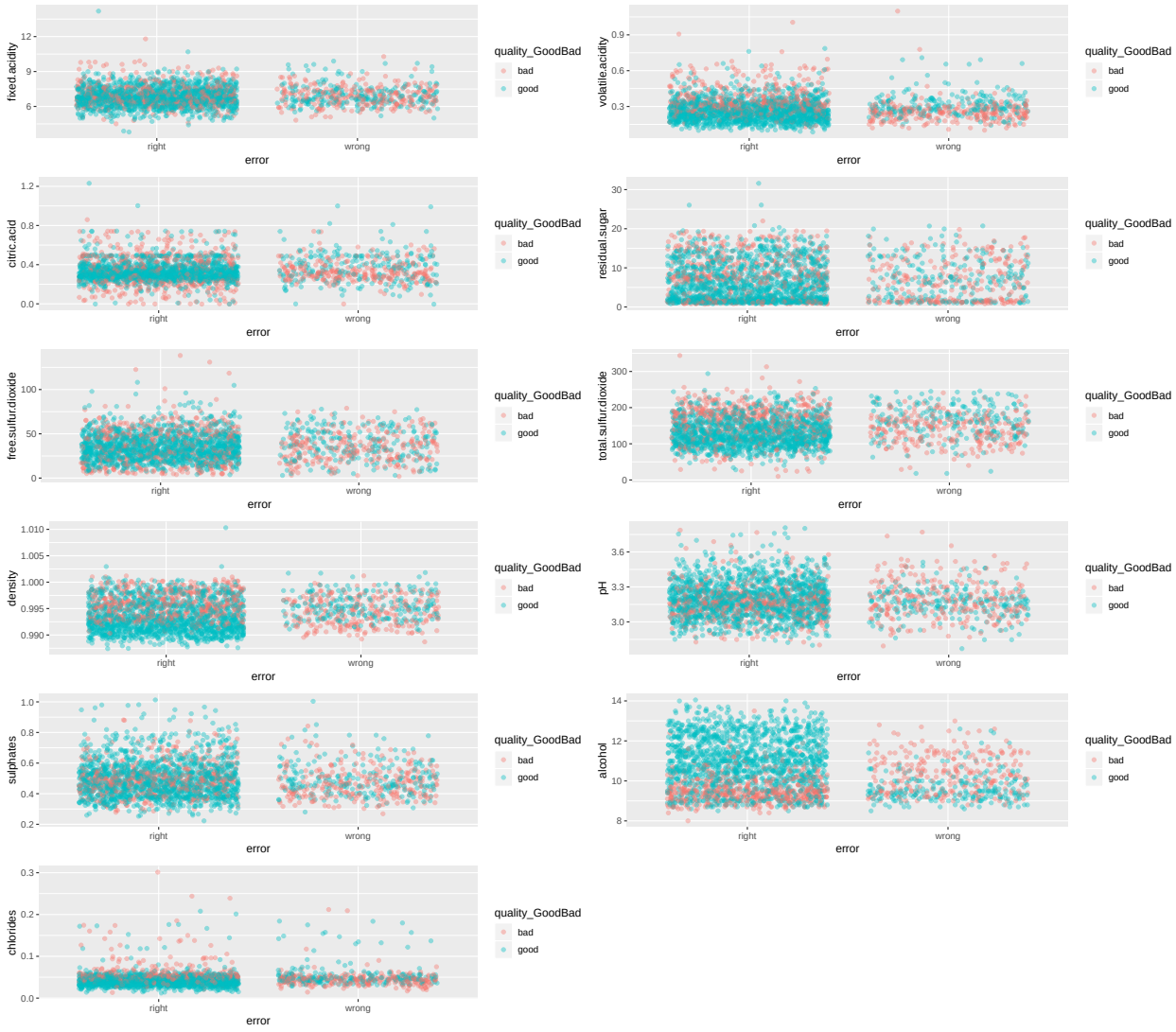
As mentioned before, the random forest predicted the quality ratings after further grouping with a good but slightly lower overall accuracy (see exact value above), which might indicate a useful prediction. However, it still did a worse job at predicting "bad" than "good" wines.

Error Investigation

We tried to further investigate where the errors occurred. Actually, the total amount of errors has increased compared to predicting three rating groups (“Average”, “Good”, “Bad”). However, we did not discover a clear pattern that would be helpful in improving our prediction.

```
## [1] 483
```

Plots Good Bad-1.bb

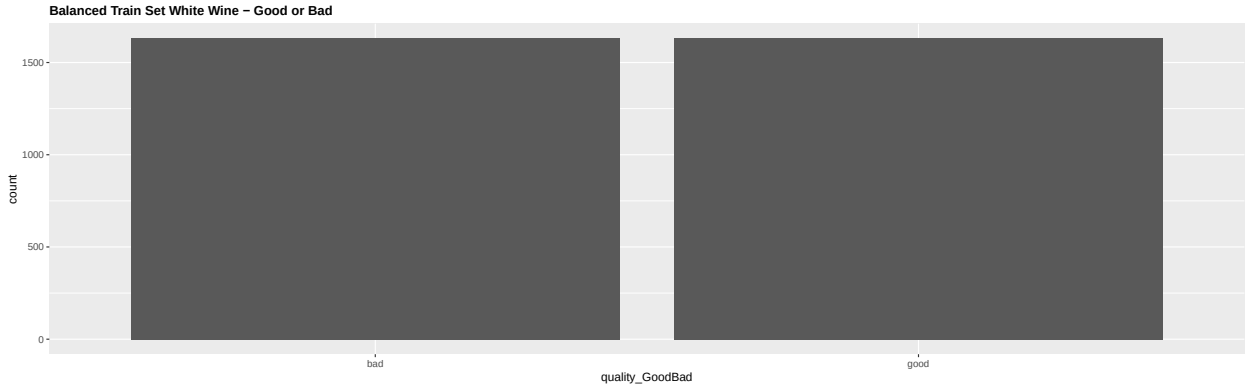


Data Set: White Wine - "Good" or "Bad"? - Attempt to Balance

Distribution and Correlation

Distribution: Many classification problems turn out to be imbalanced. This data set appears to be imbalanced too. This is not surprising due to the fact that the provider of the data set states that *"the classes are ordered and not balanced (e.g. there are much more normal wines than excellent or poor ones)"* (Source: https://archive.ics.uci.edu/ml/datasets/wine+quality). According to "Using Random Forest to Learn Imbalanced Data" by Chen, Liaw and Breiman, there are two common approaches to solve this problem: a) cost sensitive learning (assigning a high cost to the misclassification of the minority class and trying to minimize overall cost) and b) sampling techniques (down-sampling the majority class or over-sampling the minority-class). The grouping of the quality rating into two classes already created a more balanced data set. However, we are going to further balance it by over-sampling the minority class. The example with only two classes ("Good" and "Bad") was used due to the fact that many R functions only accept two classes.

Quality Good Bad balanced-1.bb



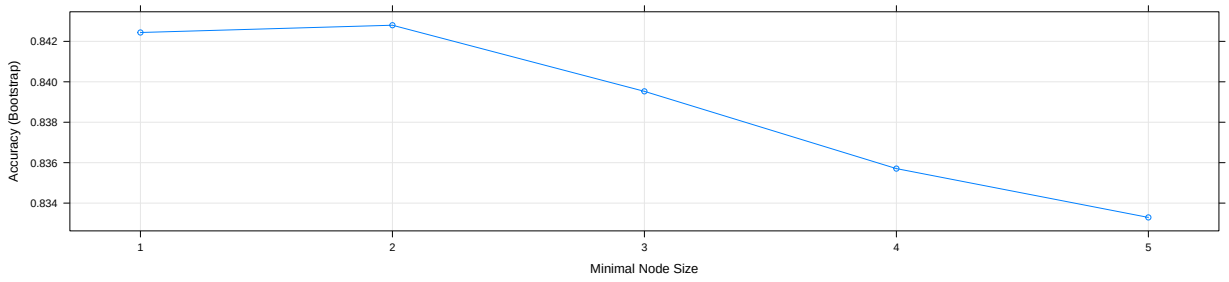
Methods/Analysis/Results

The same method was applied to the new data set (Rborist in caret train function). Except that now we tried to balance the two quality score classes of the train set. The function ovun.sample in the ROSE package was used. It *"creates possibly balanced samples by random over-sampling minority examples, under-sampling majority examples or combination of over- and under-sampling"* (Source: https://www.rdocumentation.org/packages/ROSE/versions/0.0-3/topics/ovun.sample). Furthermore, we tried to achieve improvements by assigning weights to classes. However, the weighting attempts did not result in improved accuracy and therefore were not implemented. The table "best tune" shows the combination of predFixed and minNode that resulted in the highest overall accuracy. Tuning of predFixed did not seem to cause major accuracy improvements but it increased training time substantially. The goal was once again to optimize overall accuracy, which is simply defined as the overall proportion that is predicted correctly. A change in the Rborist variable importance was discovered. However, selecting/removing variables does not seem to greatly improve overall accuracy. Changing the method to knn (k nearest neighbor) with tuning parameter k did not cause an improvement either (similar result).

	predFixed	minNode
2	0	2

Note:
best tune

Forest Quality Good Bad balanced-1.bb




```
## Rborist variable importance
##
##                       Overall
## alcohol               0.16751
## citric.acid           0.16614
## density               0.13714
## volatile.acidity      0.11738
## residual.sugar        0.10973
## chlorides             0.10840
## total.sulfur.dioxide 0.10596
## fixed.acidity         0.09561
## free.sulfur.dioxide   0.08002
## pH                    0.07587
## sulphates             0.05347
```

```
## [1] 0.7741935
```

The random forest predicted the quality after grouping into two balanced classes with a slightly lower overall accuracy (see value above), that might still indicate a useful prediction.

Confusion Matrix

We did already explain why overall accuracy can be a deceptive measure. Therefore, we constructed another confusion matrix, which illustrates the predictions and the actual values. Furthermore, it calculates overall accuracy, sensitivity and specificity.

	bad	good
bad	680	413
good	140	1216

Note:
confusion matrix

	x
Accuracy	0.7741935

Note:
overall accuracy

	x
Sensitivity	0.8292683
Specificity	0.7464702
Pos Pred Value	0.6221409
Neg Pred Value	0.8967552
Precision	0.6221409
Recall	0.8292683
F1	0.7109252
Prevalence	0.3348305
Detection Rate	0.2776644
Detection Prevalence	0.4463046
Balanced Accuracy	0.7878693

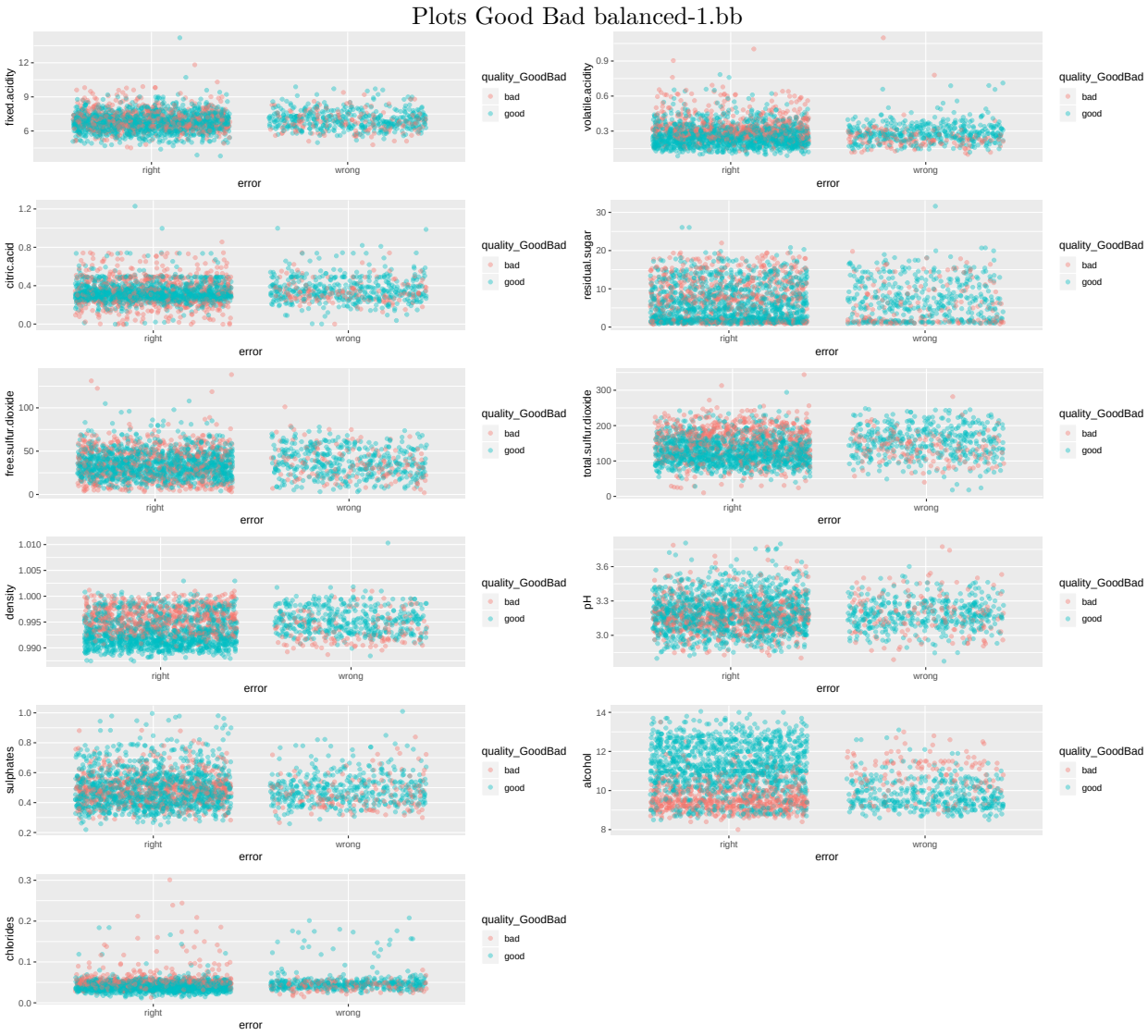
Note:
by class

As mentioned before, the random forest predicted the quality ratings after grouping into two balanced classes with a lower overall accuracy (see value above), which might still indicate a useful prediction. We managed to do a better job at predicting "bad" wines. However, we did a worse job at predicting "good" wines compared to the example with imbalanced data. As a result, the accuracy is now similar in both classes.

Error Investigation

We tried to further investigate where the errors occurred. We can see that the amount of errors increased. However, still no clear pattern visible that might help to further improve.

```
## [1] 553
```

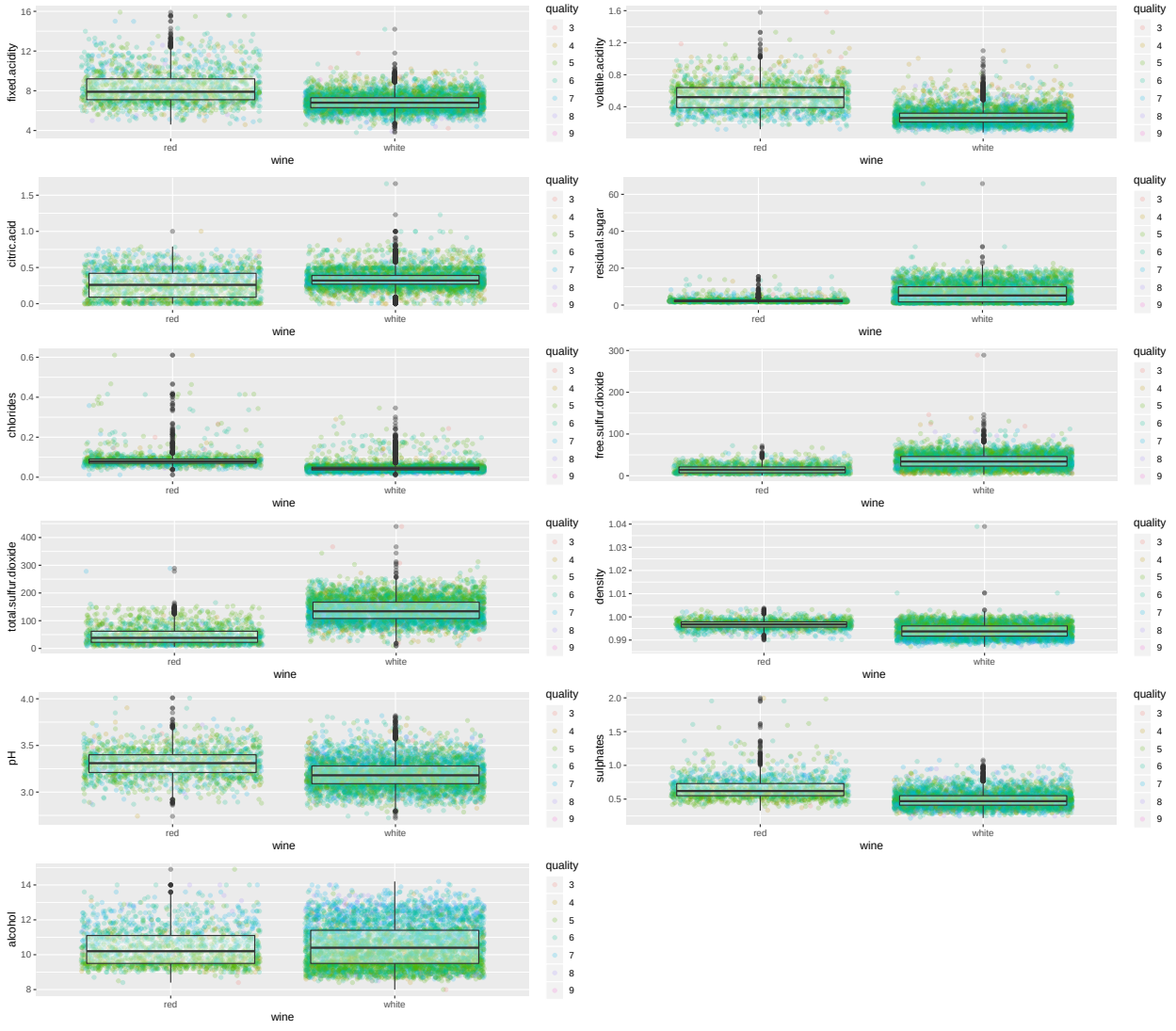


Data Set: Complete Data Set - White or Red Wine?

Data Set Structure

There are two data sets available - one for red and one for white wine. We are going to merge the two data sets. Afterwards, we are going to try to predict if a row contains a red or a white wine. An initial familiarization with the newly combined data of white and red wine was necessary. The most obvious discovery was that we made much more data available per group (which is now the wine type). There seem to be differences between the two wine types, e.g. the data set contains higher values of total sulfur dioxide for white wine than red wine and higher values of volatile acidity for red wine than white wine. Please find below the plots (jitter combined with boxplot).

Type of Wine-1.bb

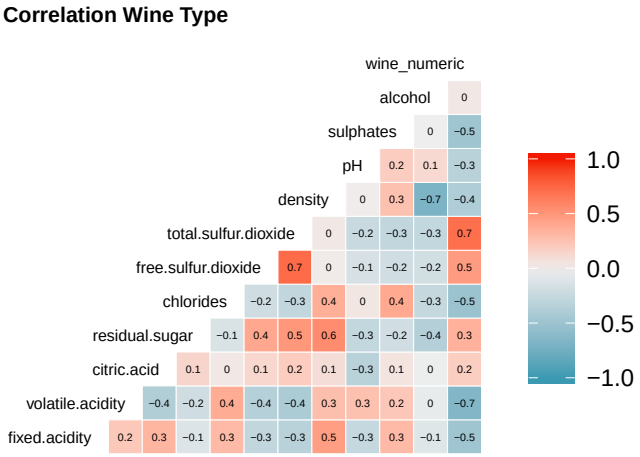


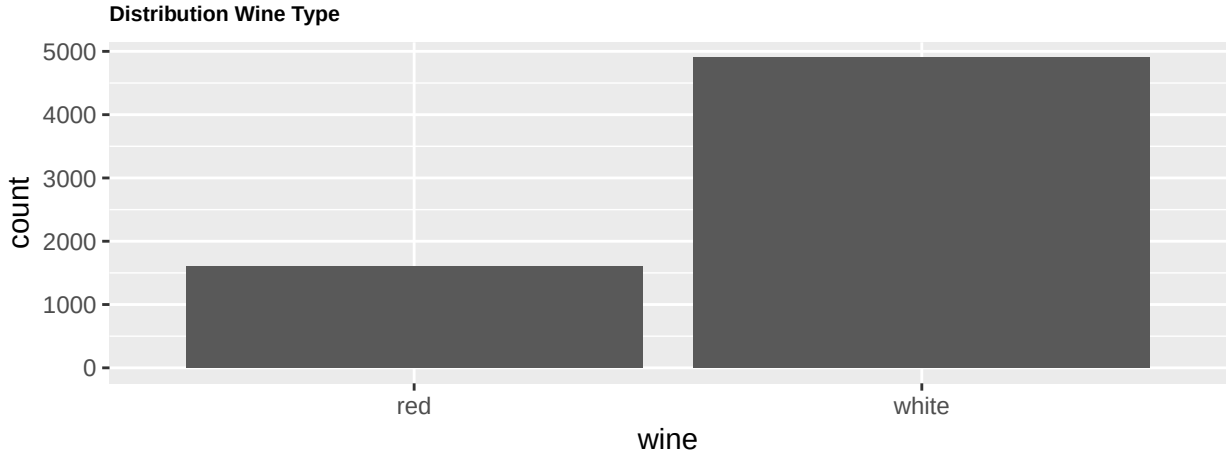
Distribution and Correlation

Distribution: The distribution shows that there is much more white wine than red wine in the data set. However, the red wine data set still contains 1599 rows.

Correlation: The correlation matrix does now show a much higher correlation between many of the wine properties and the wine types. Total sulfur dioxide shows a correlation of 0.7 and volatile acidity of -0.7. The additive sulfur dioxide is an accepted winemaking practice. It is produced in small amounts by wine yeast during alcoholic fermentation. However, most of it has been added by the winemaker at most white winemaking process steps and less liberally during red winemaking (Source: https://www.aromadictionary.com/articles/sulfurdioxide_article.html). Earlier the boxplots also indicated that the data set contains higher values of total sulfur dioxide for white wine than red wine. Volatile acidity is a problem at higher levels (limits: 1.4 g/L for red and 1.2 g/L for white) and can even smell like nail polish remover. Lower amounts can even be pleasant (e.g. fruity-smelling raspberry or cherry-like flavors). Wines that have long fermentation periods (e.g. the red wines Amarone della Valpolicella and Barolo) generally accumulate higher levels of volatile acidity (Source: https://winefolly.com/tutorial/weird-wine-flavors-and-the-science-behind-them/). Earlier the boxplots indicated that the data set contains higher values of volatile acidity for red wine than white wine. Remember that a 0 indicates no correlation, -1 a perfect negative and 1 a perfect positive correlation.

of Wine Correlation Distribution-1.bb





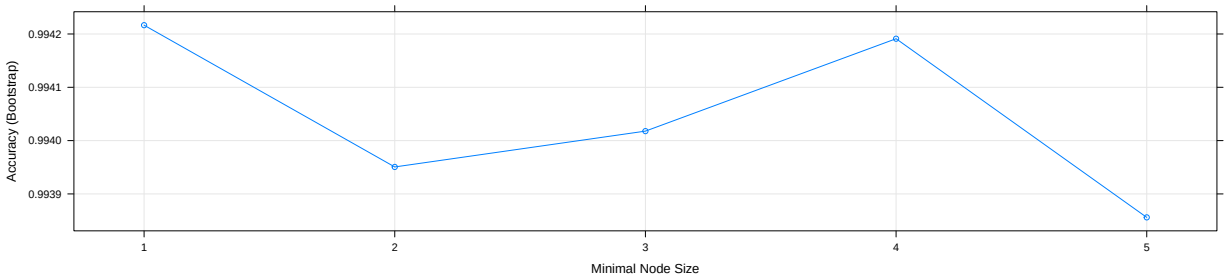
Methods/Analysis/Results

Let's try to predict if the row contains a red or a white wine. The initial goal was again to optimize overall accuracy, which is still defined as the overall proportion that is predicted correctly. The table "best tune" shows the combination of predFixed and minNode that resulted in the highest overall accuracy. The variable importance changed from training to training. However, sulfur dioxide, chlorides and volatile acidity were usually among the top values. None of the wine properties were removed.

predFixed	minNode
0	1

Note:
best tune

Forest Type of Wine-1.bb




```
## Rborist variable importance
##
##                       Overall
## chlorides            0.712496
## residual.sugar       0.475757
## total.sulfur.dioxide 0.206212
## density              0.070946
## volatile.acidity     0.050741
## pH                   0.024003
## sulphates            0.023271
## citric.acid          0.015323
## fixed.acidity        0.013008
## alcohol              0.004327
## free.sulfur.dioxide  0.004019
```

```
## [1] 0.9938443
```

The random forest managed to predict the type of wine with a very high overall accuracy (see above). This might indicate a very useful prediction.

Confusion Matrix

We did already explain why overall accuracy can be a deceptive measure. Therefore, we constructed a confusion matrix, which illustrates the predictions and the actual values. Furthermore, it calculates overall accuracy, sensitivity and specificity.

	red	white
red	787	7
white	13	2442

Note:
confusion matrix

	x
Accuracy	0.9938443

Note:
overall accuracy

	x
Sensitivity	0.9837500
Specificity	0.9971417
Pos Pred Value	0.9911839
Neg Pred Value	0.9947047
Precision	0.9911839
Recall	0.9837500
F1	0.9874529
Prevalence	0.2462296
Detection Rate	0.2422284
Detection Prevalence	0.2443829
Balanced Accuracy	0.9904458

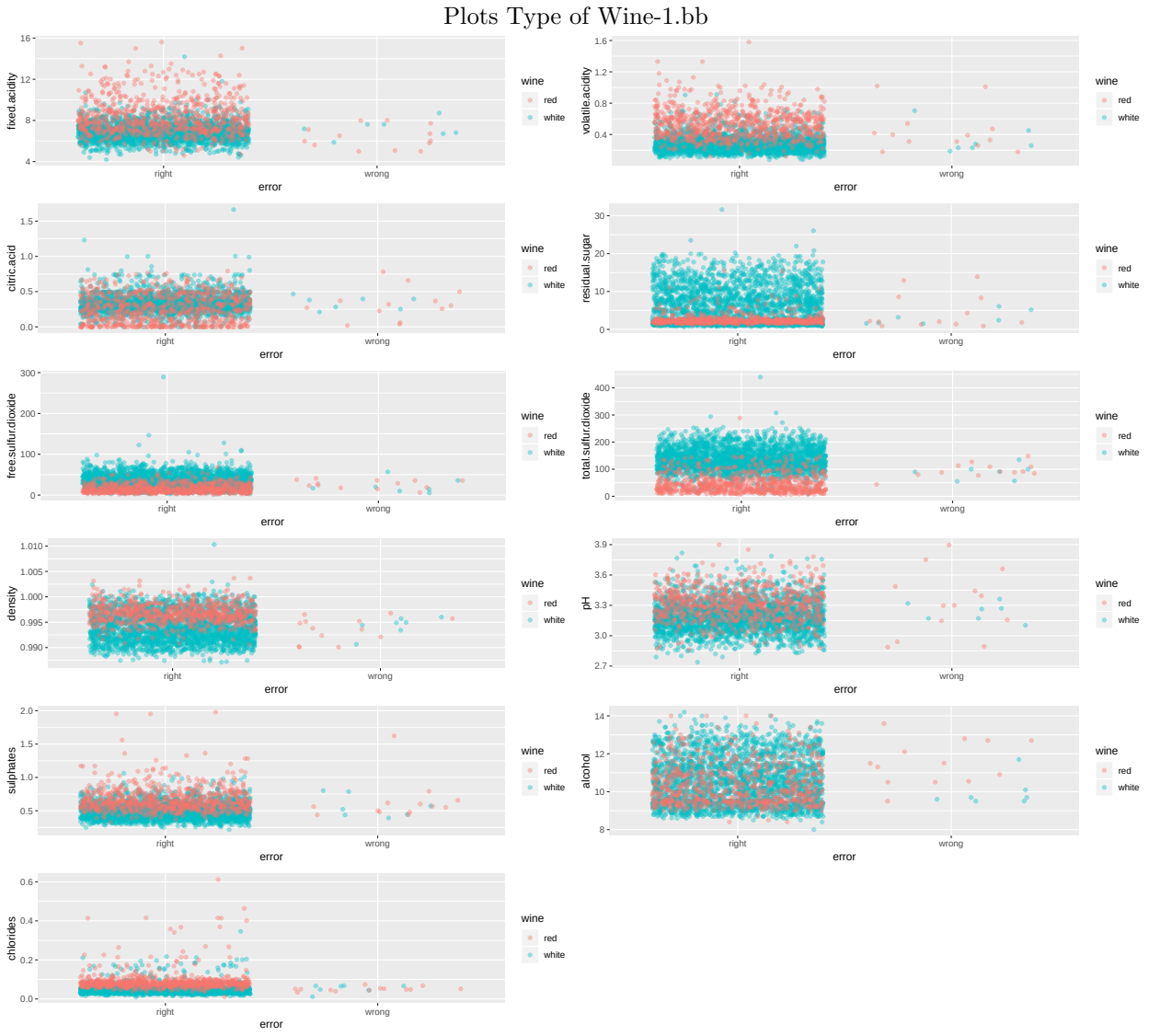
Note:
by class

The random forest predicted the type of wine with a very high overall accuracy (see exact value above), which indicates a very useful prediction. In contrary to the two earlier examples the sensitivity and specificity was high too and almost equal for the two classes.

Error Investigation

We tried to further investigate where the errors occurred. We only made a few mistakes. However, we did not manage to find a clear pattern that describes how to get rid of these.

```
## [1] 20
```



Conclusion

Random Forest 1 / White Wine - Rating on a Scale from 0 to 10?

The random forest method was able to predict the quality scores ranging from 1 to 10 with an overall accuracy of approximately 63 per cent (precise value for accuracy might slightly differ from training to training). Furthermore, it was mainly good at predicting "average" wines and did a poor job at detecting "bad" and "good" wines. Most likely a winemaker would want to find the recipe for the "good" wine and avoid the "bad" wine. Therefore, the overall usefulness seems limited.

Random Forest 2 / White Wine - "Good", "Average" or "Bad"?

After grouping into three categories such as "good", "average" and "bad", the same random forest achieved an accuracy of approximately 85 per cent (exact accuracy might slightly differ from training to training). This seems already much better. However, it was mainly good at predicting "average" wine and still did a bad job at predicting "good" wine and a very bad job at predicting "bad" wine. This is especially unfortunate due to the fact, that the few bad wines in the data set were usually overrated as "average".

Random Forest 3 / White Wine - "Good" or "Bad"?

Further grouping into only two categories "good" and "bad" resulted in an accuracy of approximately 82 per cent with a much better sensitivity and specificity (precise accuracy might slightly differ from training to training). However, we are still better at predicting "good" than "bad" and we lose lots of information by aggregating eleven quality scores into two. Furthermore, we still get many predictions wrong. Further balancing of the data set by over-sampling the "bad" class of the wine training set did not result in an improved overall accuracy (approximately 77 % / exact accuracy might slightly differ from training to training) but it resulted in an almost equal accuracy for the two classes. I guess it depends on the prediction's goal if the balanced or the imbalanced data set should be considered as the preferred choice.

Random Forest 4 / White Wine - Possible Limitations

The physicochemical properties do not seem to be clearly distinguishable from one quality rating to the other. A possible reason might be the lack of data for all the bad and good ratings. The data set mainly contains average wines. Therefore, it might be necessary to collect further data to balance the data set. Additionally, the exact meaning of the ratings ranging from 1 to 10 was unknown. Therefore, it was hard to capture the logic in corresponding groups (e.g. where does "average" end and "bad" start). Lastly, wine quality seems to be a challenging topic, e.g. volatile acidity might even add complexity and interest to many wines but in high quantities it can be disturbing (similar to the smell of vinegar or nail polish).

Random Forest 5 / Complete Data Set - White or Red Wine?

The random forest seems to perform very well in predicting "red" and "white" wine, where the physicochemical properties seem to be more distinct and the available data larger. We achieved an overall accuracy, sensitivity and specificity of approximately 99 per cent (exact value might slightly differ from training to training). However, I do not know if it is useful to predict the type of wine from 11 wine properties. There are probably easier methods to do that (e.g. looking at the colour). Nevertheless, we still might be able to learn something about the differences of red and white wine.
