White Wine Quality

Capstone: Choose Your Own

Luis D. Torres

12/28/2020

Introduction

In this work, three machine learning techniques are used to determine dependency of white wine quality on eleven wine characteristic. Machine learning techniques are compared by calculating the level of accuracy of the model.

The wine quality dataset is used in this report. The wine dataset is a collection of white and red wines. The dataset is publicly available at https://archive.ics.uci.edu/ml/datasets/wine+quality.

This report uses only the white wine data as it includes more observations.

This report is structured in three sections. The next section explains the methods and analysis used including data cleaning, data exploration and visualization, and the modelling approach. The results section presents the modelling results and discusses the model performance. The final section gives a brief summary of the report, its limitations and future work.

Methodology

Datasets

Wine quality dataset is used in this report. Wine dataset is a collection of white and red wines. This report uses only the white wine data with 4898 observations.

The white wine quality dataset can be download at https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-white.csv

A summary of the variables included in this dataset is provided below:

The dataset was divided in two main subsets: 1. Training: 80% of the white wine quality dataset. Total observations of 3,918 2. Validation: 20% of white wine quality dataset. Total observations of 980

All models were trained on the **Training** dataset. To facilitate model comparison and reduce the risk of over-training, **Training** was further partitioned into *train* (80% of Training) and *test* (20% of Training).

The **Validation** dataset was not used for training purposes. This set was used only for full validation. This implies testing the final and best performing algorithm.

Variables (train set)

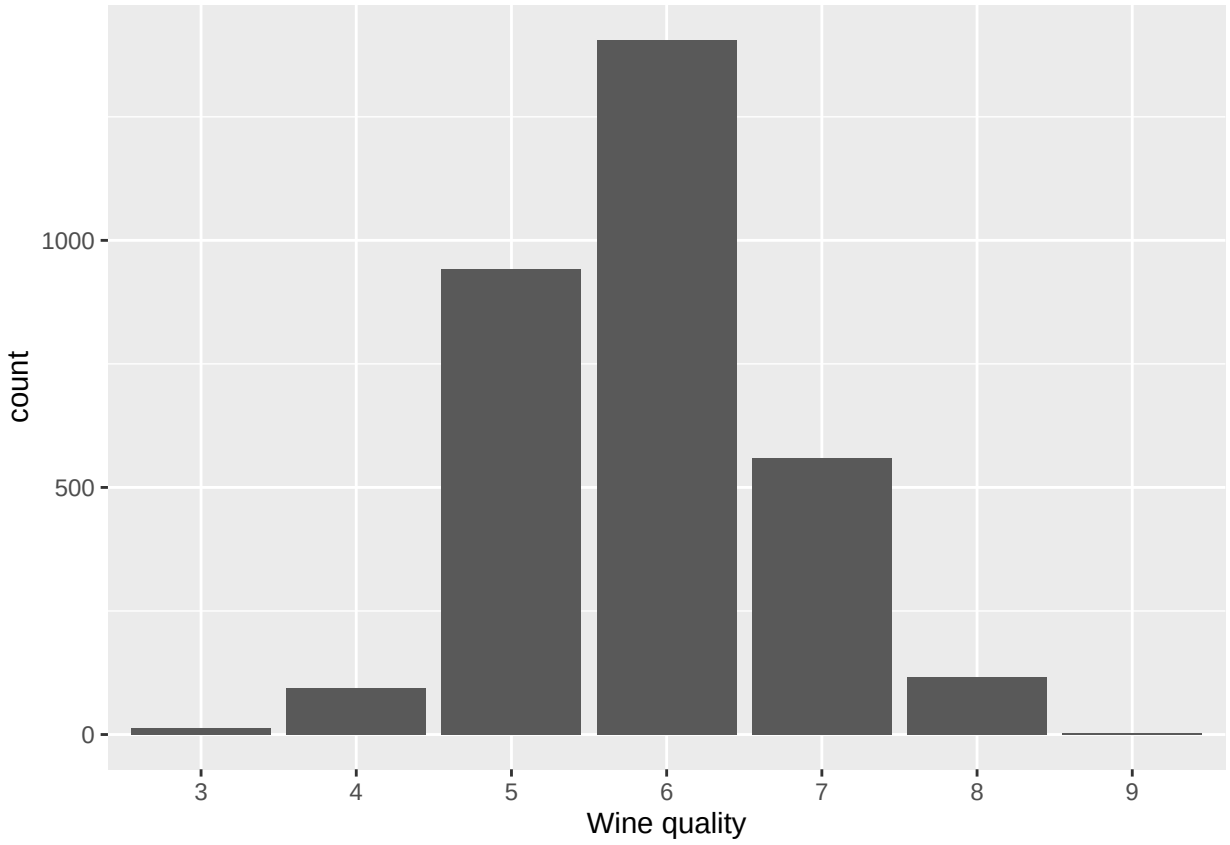
The summary of the variables included in the train dataset are summarised below:

	n	mean	sd	median	min	max
fixed.acidity	3133	6.86	0.83	6.80	3.90	10.70
volatile.acidity	3133	0.28	0.10	0.26	0.08	1.00
citric.acid	3133	0.34	0.12	0.32	0.00	1.66
residual.sugar	3133	6.44	5.13	5.20	0.60	65.80
chlorides	3133	0.05	0.02	0.04	0.01	0.35
free.sulfur.dioxide	3133	35.81	17.09	34.00	3.00	289.00
total.sulfur.dioxide	3133	139.39	42.72	135.00	10.00	440.00
density	3133	0.99	0.00	0.99	0.99	1.04
pH	3133	3.19	0.15	3.18	2.79	3.82
sulphates	3133	0.49	0.11	0.48	0.23	1.08
alcohol	3133	10.50	1.24	10.40	8.00	14.20
quality	3133	5.88	0.88	6.00	3.00	9.00

Outcome variable: Quality

The wine *quality* rating is an ordinal variable based on a sensory test carried out by at least three sommeliers and scaled in 10 quality classes from **0** - *very bad* to **10** - *very excellent*.

By exploring the train dataset, wine quality do not include values of 1, 2 and 10. Most values are around 5, 6 and 7 as showed on the graph below:



Features (predictors)

The wine quality dataset included 11 continuous variables that can be used as predictors of wine quality ratings: *fixed acidity*, *volatile acidity*, *citric acid*, *residual sugar*, *chlorides*, *free sulphur dioxide*, *total sulphur dioxide*, *density*, *pH*, *sulphates*, *alcohol*.

Preprocessing was implemented on the set of predictors. Variables with zero variability can represent a problem for the machine learning algorithms. Therefore, variability was tested for all predictors. As showed on the table below, zero variability is not a problem in this set of variables so all variables can be included in the analysis:

	zeroVar
fixed.acidity	FALSE
volatile.acidity	FALSE
citric.acid	FALSE
residual.sugar	FALSE
chlorides	FALSE
free.sulfur.dioxide	FALSE
total.sulfur.dioxide	FALSE
density	FALSE
pH	FALSE
sulphates	FALSE
alcohol	FALSE

Data analysis

Analysis were performed using R version 4.0.3 and RStudio version 1.3.959. The following R packages were used: *tidyverse*, *data.table*, and *psych* for data manipulation and visualisation; and the *caret* for building three machine learning algorithms: *k-nearest neighbours*, *decision trees*, and *random forests*.

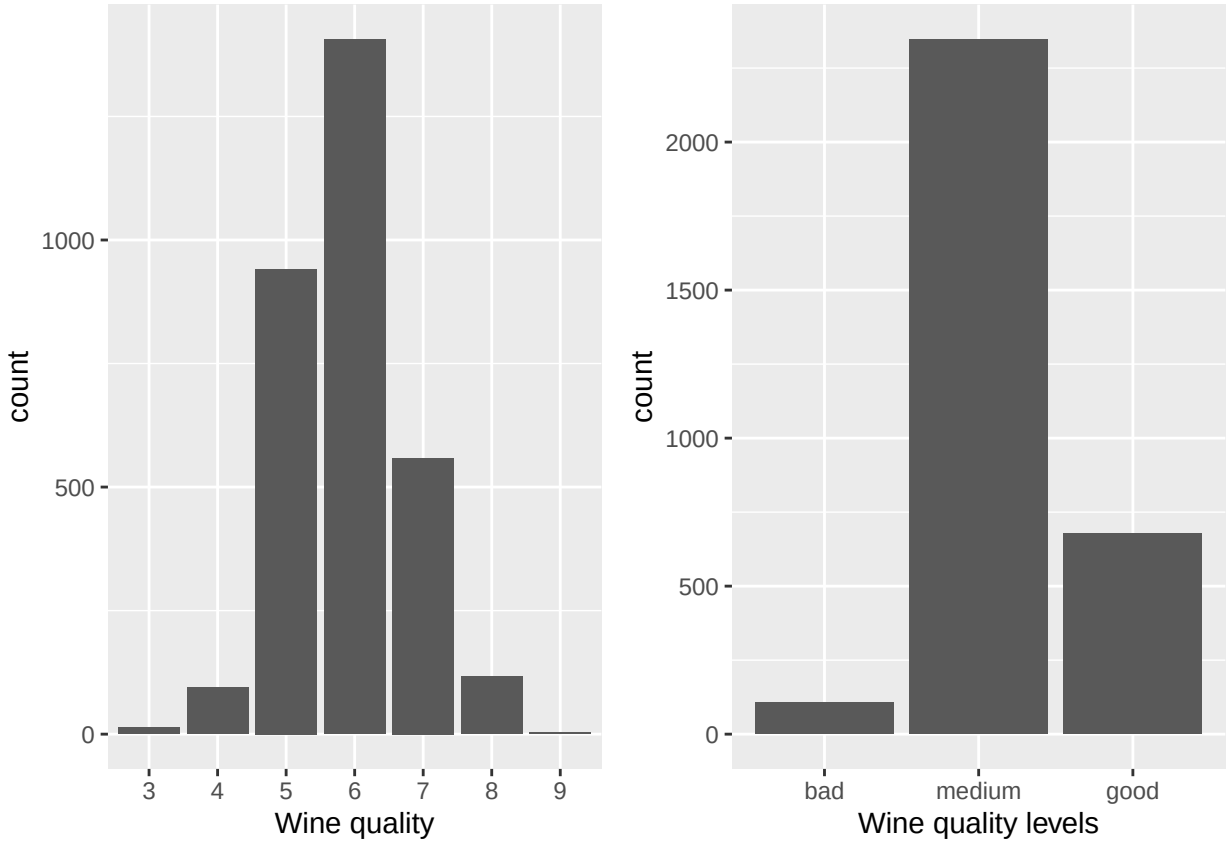
This report uses *overall accuracy* as the main indicator to compare models and their predictive value. This is the overall proportion of cases that were correctly predicted. The model should aim at an *overall accuracy* value closer to 1.

Bootstrap-based cross validation is used to optimise the machine learning algorithm parameters. This facilitates choosing the parameters with the highest *overall accuracy* while reducing the risk of over-training.

Data transformations and model fitting

The outcome variable (quality) was used in two different ways: 1. Raw (quality): no transformation was implemented 2. Transformed (levels): raw ratings were transformed into three levels as follows: * Bad = ratings 1 to 4 * Medium = ratings 5 and 6 * Good = ratings 7 to 10 A new variable called “levels” was added to all datasets.

The distribution of observations for the raw and transformed outcome variables in showed in the figure below:



The three machine learning algorithms were fitted to the raw and transformed outcome variable.

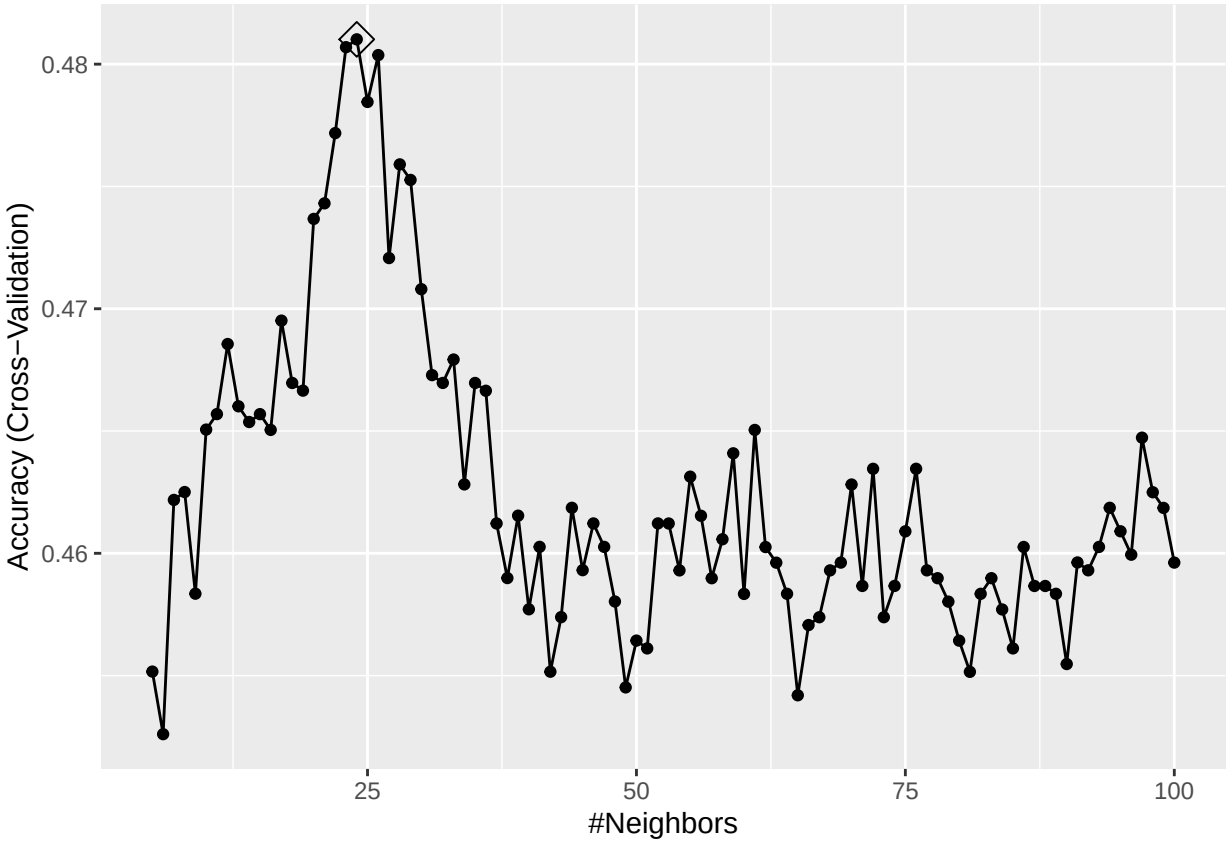
Results

Fitting algorithms using the raw (quality) outcome variable

Model 1 raw: K-nearest neighbours

The first algorithm fitted to the train dataset is the k-nearest neighbours. In order to optimise the parameter k, a set of 5 to 100 k-neighbours were trained.

The figure below shows the results of this process:



As shown, he parameter k that maximised the estimated accuracy is:

```
##     k
## 20 24
```

By fitting this optimised parameter, the model improves the accuracy to:

```
## Accuracy
##    0.462
```

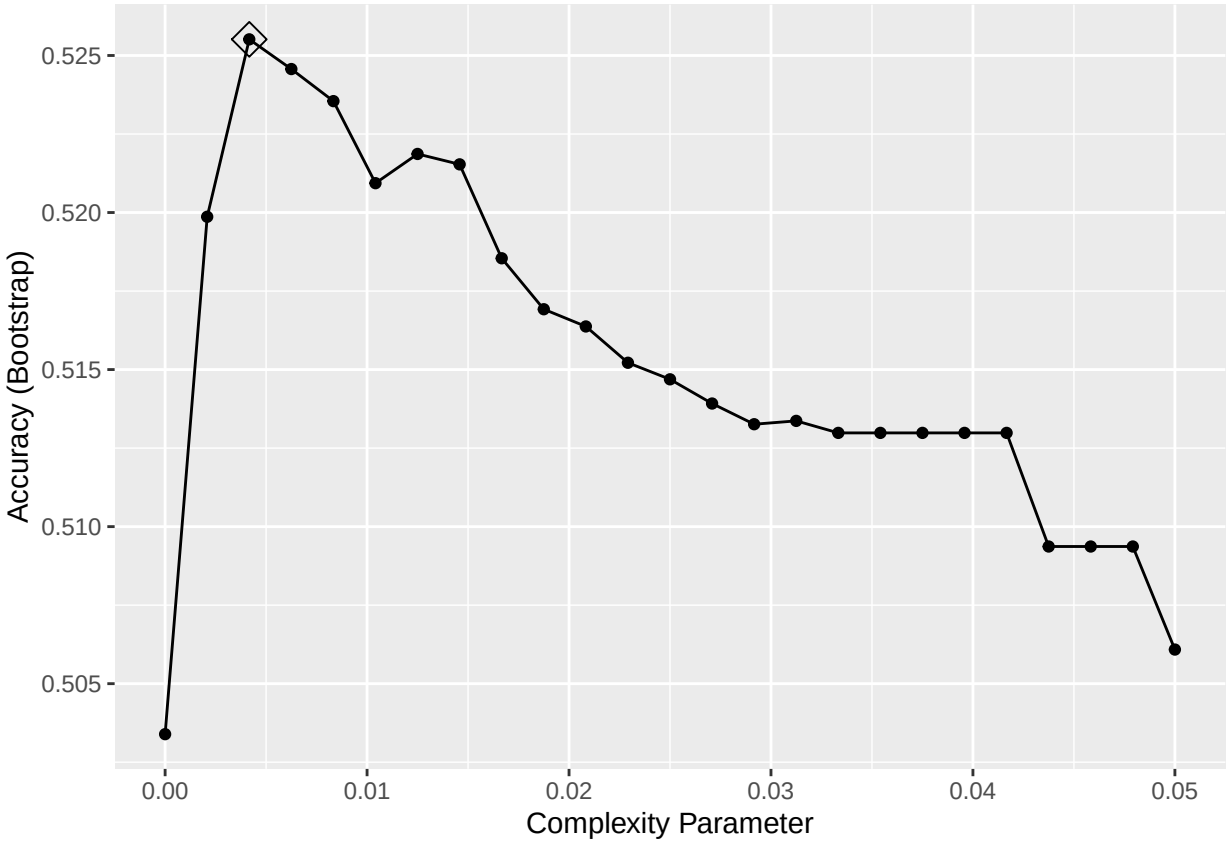
The resulted accuracy does no achieve even 50% which means that the k-nearest neighbours does worse than guessing.

Model 2 raw: Decision trees

Decision trees can model human decision processes and do not require use of dummy predictors for categorical variables as for traditional regression methods.

Decision trees create partitions recursively starting with one partition or the entire predictor space. They form predictions by calculating which class is the most common among the training set observations within the partition.

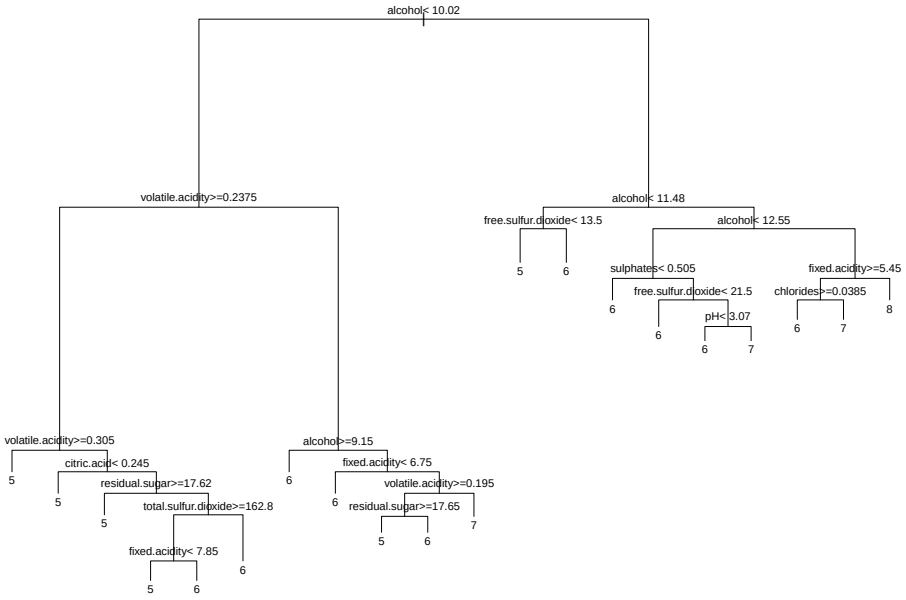
In order to optimise the numbers of partitions to be created, the best complexity parameter (cp) can be chosen. By using cross validation, the optimised *cp* for this algorithm can be selected from the figure below:



The main idea is that accuracy must improve by a factor of *cp* for the new partition to be added. The optimised *cp* is:

```
## [1] 0.00417
```

The final model using this parameters can be graphically displayed in the figure below. As showed on the figure, the first split is made at alcohol lower than 10.2. The two resulting new partitions are split at volatile acidity higher or equal than 0.2375 and alcohol lower than 11.48, respectively, and so on.



By fitting this optimised algorithm, the model improves the accuracy to:

```
## Accuracy
##    0.541
```

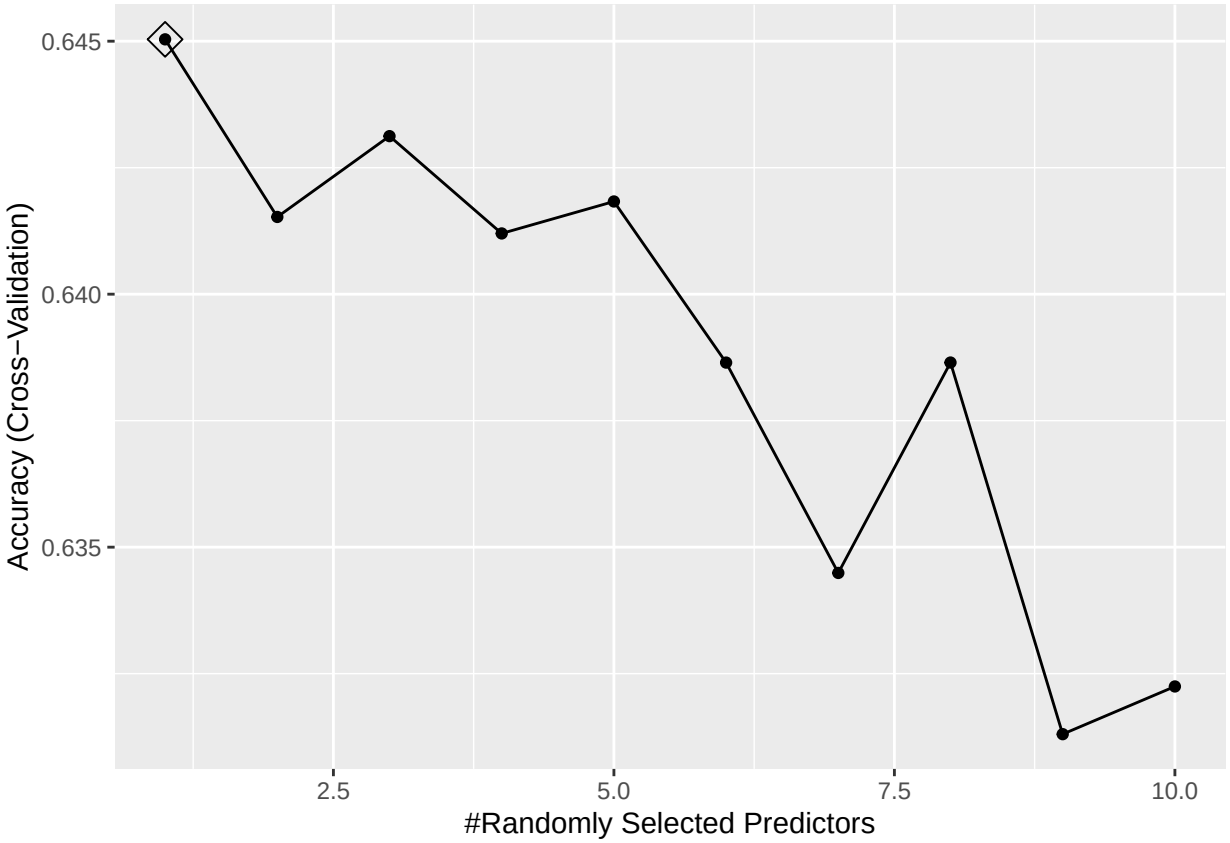
This is better than k-nearest neighbours, but not much improvement from a simple guessing.

Model 3 raw: Random forests

Decision trees is rarely the best performing method in terms of accuracy since it is not very flexible and is highly unstable to changes in training data. Random forests improve on several of these shortcomings.The goal is to improve prediction performance and reduce instability by averaging multiple decision trees.

The first step is bootstrap aggregation. The general idea is to generate many predictors, decision trees, and then forming a final prediction based on the average prediction of all these trees. Therefore, the bootstrap makes the individual trees randomly different, and the combination of trees is the forest.

In the first step the parameter that controls the minimum number of data points in the nodes of the tree is optimised using cross validation. The figure below shows the result of this process:



The optimised parameter is:

```
##   mtry
## 1    1
```

By fitting the model with this parameter, accuracy improves to:

```
## Accuracy
##    0.652
```

By examining the variable importance, it is possible to see how often a predictor is used in the individual trees. As shown on the table below, alcohol, density and volatile acidity receive the highest values:

	MeanDecreaseGini
fixed.acidity	159
volatile.acidity	218
citric.acid	174
residual.sugar	187
chlorides	178
free.sulfur.dioxide	194
total.sulfur.dioxide	195
density	219
pH	179
sulphates	169
alcohol	239

Random forest performs the best compare to all the previous algorithms. However, accuracy remains just over 65% which is low.

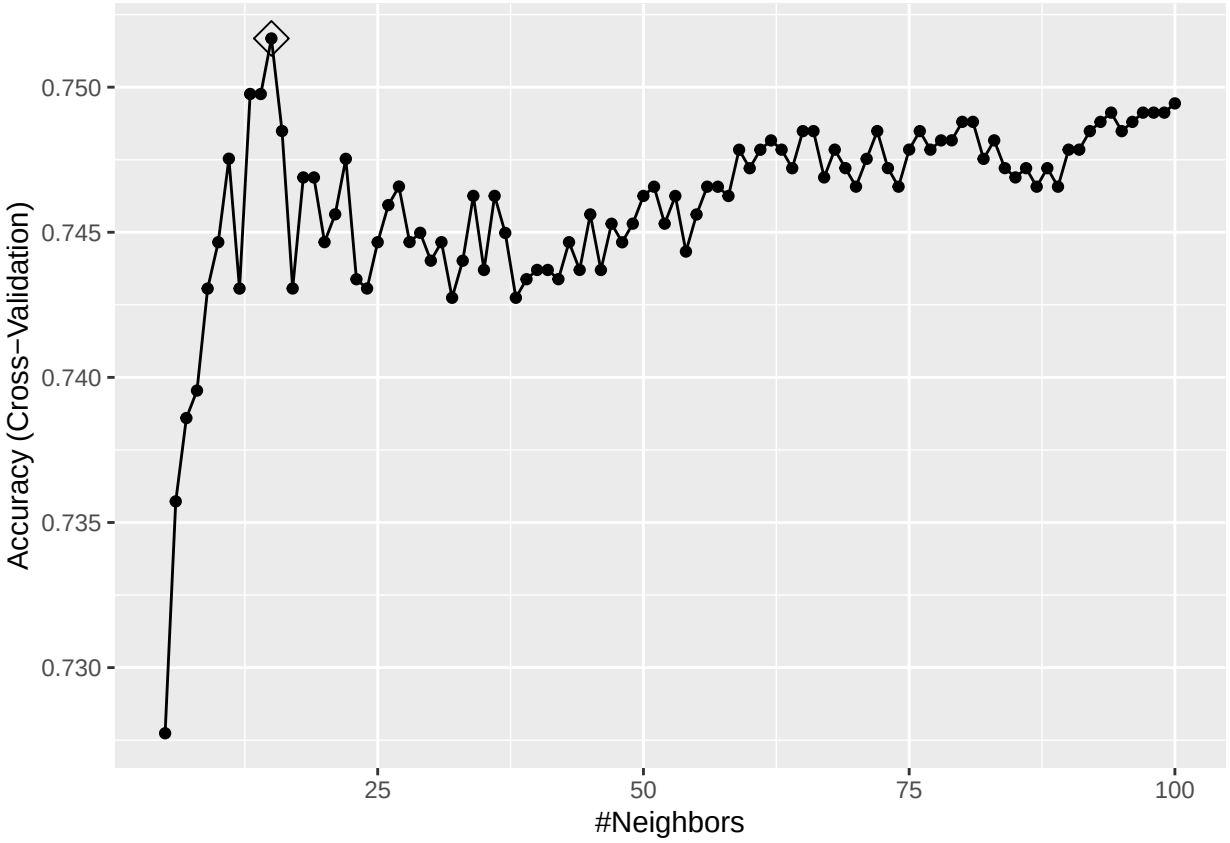
Fitting algorithms using the transformed (levels) outcome variable

The same algorithms were fitted to the transformed outcome variable (levels). Three levels are considered: * Bad = ratings 1 to 4 * Medium = ratings 5 and 6 * Good = ratings 7 to 10

The same procedure as in the previous section was implemented in each case.

Model 1 levels: k-nearest neighbours

In order to optimise the parameter k, a set of 5 to 100 k-neighbours were trained and the result is displayed in the figure below:



As shown, he parameter k that maximised the estimated accuracy is:

```
##     k
## 11 15
```

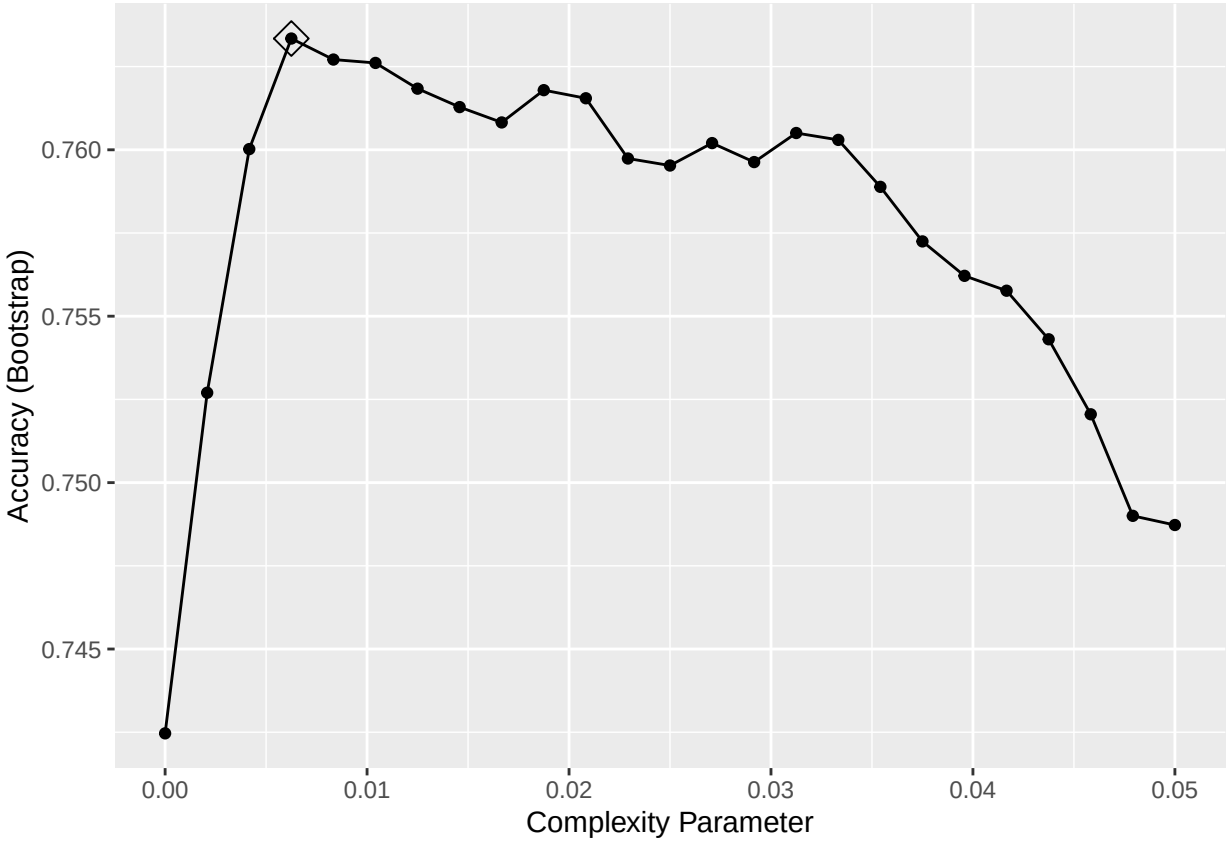
By fitting this optimised parameter, the model improves the accuracy to:

```
## Accuracy
##    0.753
```

This is a improvement of almost 30% considered the first implementation of the k-nearest neighbours algorithm.

Model 2 levels: Decision trees

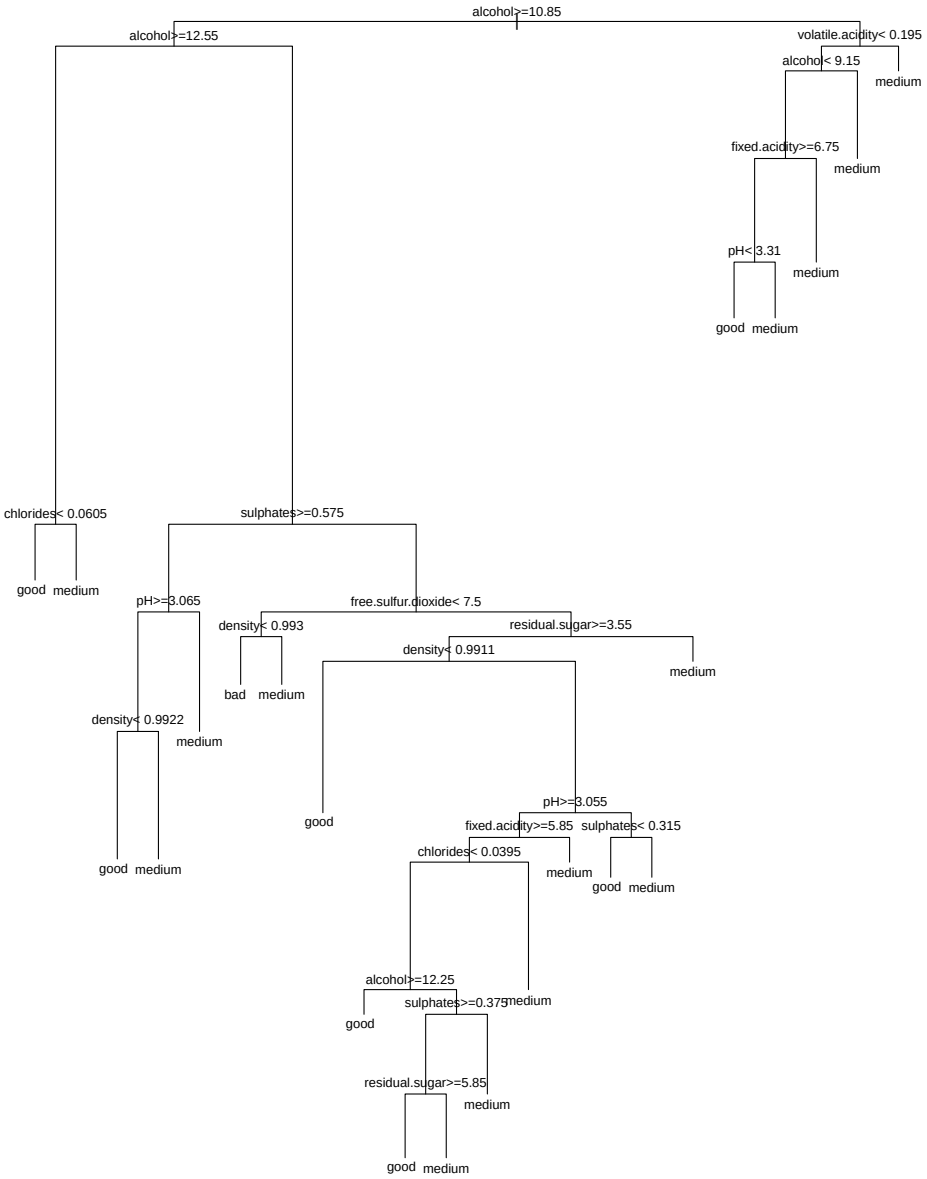
Following the first implementation of decision trees, the optimised *cp* can be selected by using cross validation. The figure below shows the result of this procedure:



The optimised *cp* is:

```
## [1] 0.00625
```

The final model using this parameters can be graphically displayed in the figure below. As showed on the figure, the first split is made at alcohol higher or equal than 10.85. The two resulting new partitions are split at volatile acidity lower than 0.195 and alcohol higher or equal than 12.55, respectively, and so on.



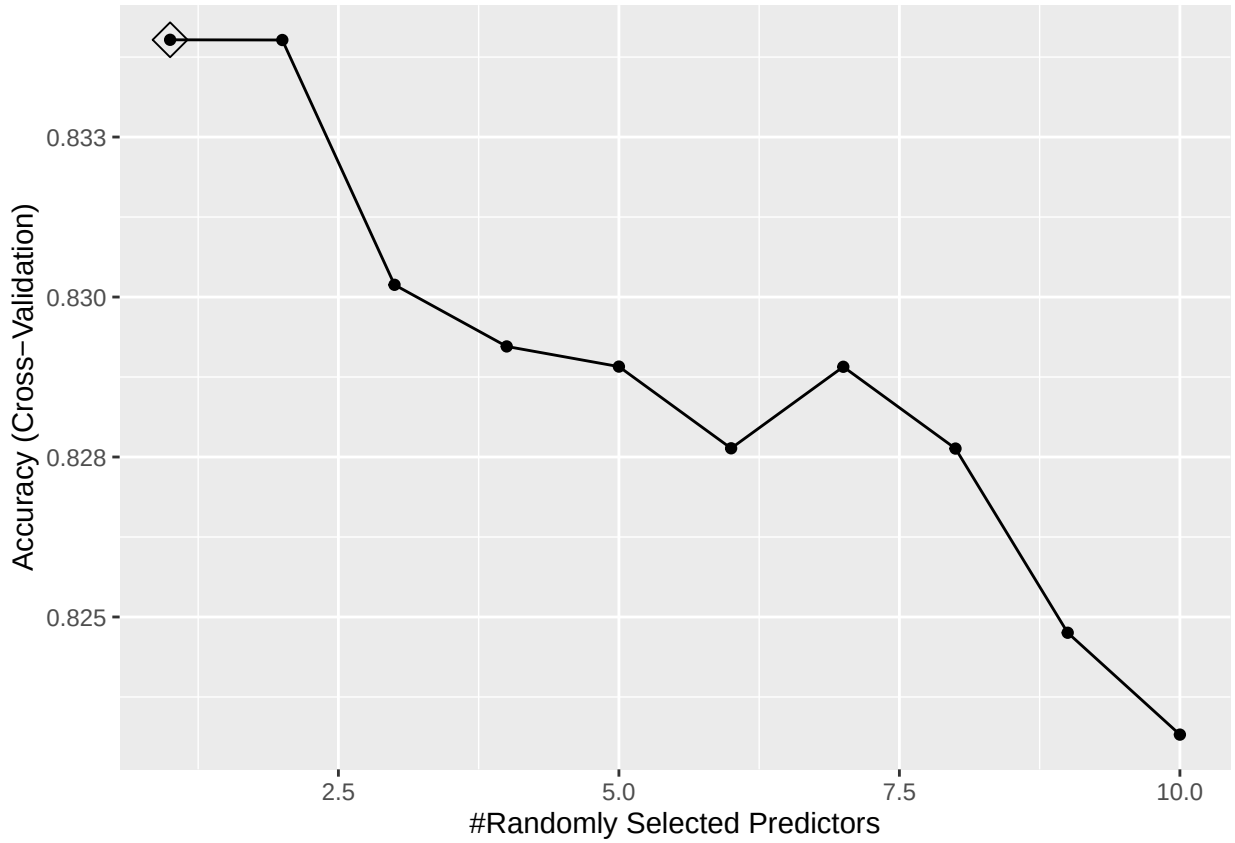
By fitting this optimised algorithm, the model improves the accuracy to:

```
## Accuracy
##    0.782
```

This is better than the initial decision trees accuracy and slightly higher than k-nearest neighbours.

Model 3 levels: Random forests

The parameter that controls the minimum number of data points in the nodes of the tree is optimised using cross validation. The figure below shows the result of this process:



The optimised parameter is:

```
##   mtry
## 1    1
```

By fitting the model with this parameter, accuracy improves considerably to:

```
## Accuracy
##    0.834
```

By examining the variable importance, it is possible to see how often a predictor is used in the individual trees.As shown on the table below, alcohol, density and residual sugar receive the highest values. Residual sugar takes the place of volatile acidity from the first implementation of random forests.

	MeanDecreaseGini
fixed.acidity	85.9
volatile.acidity	108.0
citric.acid	86.0
residual.sugar	110.6
chlorides	107.2
free.sulfur.dioxide	112.6
total.sulfur.dioxide	102.9
density	140.5
pH	102.0
sulphates	98.7
alcohol	170.2

Validation with the best performing model

Random forest in the second condition where wine quality has been transformed into 3 levels showed the best predictive performance.

The final random forest algorithm is tested in the validation dataset in order to confirm its accuracy. By doing so, the achieve accuracy is:

```
## Accuracy
##    0.823
```

The resulted accuracy is over 80% and almost the same as the one achieved during training.

Summary and conclusions

In this work, *k-nearest neighbours*, *decision trees*, and *random forests* were used to determine dependency of white wine quality on eleven wine characteristic.

These three algorithms were fitted to the train dataset using wine quality as the outcome variable. Wine quality was used in its *raw* form (1 to 10 quality ratings) and transformed intro *levels* (bad, medium, good).

The table below summarises the results of fitting models to the outcome variable in its two forms: raw and levels.

Method	Accuracy
Model 1 raw: k-nearest neighbours	0.462
Model 2 raw: Decision tree	0.541
Model 3 raw: Random forest	0.652
Model 1 levels: k-nearest neighbours	0.753
Model 2 levels: Decision tree	0.782
Model 3 levels: Random forest	0.834
Best performing model validation	0.823

By using the raw form of the outcome variable, algorithms achieved a maximum accuracy of about 66%, being random forests the best performing algorithm.

By transforming the raw outcome variable intro 3 levels, all algorithms achieved a better performance with a maximum accuracy of about 83%. Random forests was again the best performing algorithm.

The final random forest algorithm was tested in the validation dataset in order to confirm its predictive value. The resulted accuracy is close to the one achieved during the training process confirming random forest as the best algorithm.

Limitations and future research

The analyses presented in this report have several limitations. The outcome variable did not include very low (1 and 2) and very high (10) ratings. In addition, most observations were concentrated at middle ratings (5, 6 and 7). This may have an impact on the algorithms as prevalence can reduce the predictive capacity by reducing or inflating it. Also, predictors differed greatly regarding their measurement values. This may be inflating the real predictive value of such a predictors.

Besides dealing with these limitations, future inquiries could explore this dataset in three ways. First, future analyses can consider the outcome variable as a continuous variable and calculate the root-mean-square error (RMSE). This can facilitate features selection. For example, it could start with a linear regression to identify the most important predictors and delete the ones which do not have predictive value.

Second, future studies can also apply dimension reduction to optimise predictors. This could reduce the impact of collinearity between predictors. This is an aspect which was not explored in the present analysis.

Finally, a more focused approached could be taken by transforming the outcome variable into two categories such as good and bad or high and low quality. By doing so, the analysis would be more effective at predicting what makes a good quality wine.