Wine Quality Prediction

Yun-Chung Pan (yp392)

1. Objective

In this project, we analyzed data of red and white variants from Portuguese Vinho Verde wine to identify chemical properties that would influence the quality of wines by using machine learning algorithm in R programming. For our entire analysis, we will first look into the red wine dataset, conduct the same method in white wine data, and compare the results in two different wines.

2. Exploratory Data Analysis (EDA)

Before making data visualization, let us check some general information of raw data 'winequality-red.csv' and 'winequality-white.csv' to ensure the quality of our data analysis further on. Both data frames include 11 numeric variables and one categorical variable, with 1,599 and 4,898 observations in red and white wine datasets respectively. The 11 quantitative variables are the physicochemical properties of wines, ranging from acidity, density, and content of different chemical compounds, and they will be used as predictors in the machine-learning model. *Quality* is our target variable in this project, which classifies the wine quality using integers from 0 to 10. Using is.null( ) function, we confirmed that there are no null values in both datasets.

(1) Histogram of target variable: *quality*

Though based on definition there are eleven levels of classification in wine quality, we can see from the histograms (Fig. 1-1, 1-2) that, for both red and white wine dataset, the majority of observations lies in the middle classes (5 to 7), and there are very few data in the extremes on both sides. Due to this limitation, we can expect that it would be difficult to achieve decent accuracy in low and high quality in our

prediction model.

(2) Distribution of predictors

Here we visualized the distribution of each predictor to realize skewness and outliers in the data. In red wine dataset (Fig. 2-1-1~2-1-11), variables such as *residual sugar*, *chlorides*, and *sulphates* have highly positive skewness due to some extreme outliers; *density* and *pH* are normally distributed with fewer outliers. In white wine dataset (Fig. 2-2-1~2-2-11), except *pH* that is normally distributed, most predictors are positively skewed due to the presence of extreme outliers on the right sides.

(3) boxplot between predictors and quality variable

We used boxplots to initially analyze the correlation between each predictor and the target variable. In the red wine dataset (Fig. 3-1-1~3-1-11), we can see the upward trends in the *citric acid*, *sulfate*, and *alcohol* as quality gets higher, indicating the positive correlation in each other. On the other hand, variables such as *volatile acidity*, *density*, and *pH* show downward trends as the level of quality increases. In some sense, it is reasonable to see density and alcohol having inverse trends since from a chemistry perspective, wine is similar to an alcohol-water solution, and the density of alcohol (~0.8 g/mL) is lower than water (1.0 g/mL). In the white wine part (Fig. 3-2-1~3-2-11), we can see the upward trends in the *alcohol* and *pH* as quality gets higher, and the downward trends in *chlorides* and *density* as the level of quality increases. The visual observations described above can be verified by the correlation matrices in Fig. 4-1 and 4-2. It is interesting that the relationship between each predictor and the target variable are quite different in the two types of wines, and that in the white wine dataset there are more boxplots that do not provide obvious trends for us to observe visually than the red one. Perhaps this is what machine learning comes for, to provide results that people cannot easily notice by simple observation!

3. Data Preprocessing

From the EDA section we noticed the limitation that in two datasets there are few data in both extremes, which may be problematic when predicting high and low levels of wine quality. To alleviate the problem and to provide a more meaningful categories for human beings, we redefined our target variable. Based on the frequency to the levels of quality in the datasets, we introduced a new variable called *rating*, which maps the quality to "low" (0-4), "mediocre" (5-6), and "high" (7-10). Here we employed five different machine-learning methods to analyze the data, and compared the performance in each algorithm afterwards.

The modelling dataset is split into training and testing sets by assigning 80% of data points to the former and the remaining 20% to the latter. To maintain reproducibility, set.seed(10) is used in the sampling process. All the parameters in the machine learning algorithms are tuned based on **accuracy** using 5 repeats of 5-fold cross validation in caret package.

(1) Logistic Regression

As we classified the quality into three levels, here we addressed a multinomial logistic regression. Same as binary logistic regression, it uses maximum likelihood estimation to evaluate the probability of categorical membership. In the R environment, we employed multinom( ) from *nnet* to build this model.

Unlike binary logistic regression, assigning reference level is required in multinomial logistic regression. Using relevel( ), we defined the "low" class as the reference in the regression, which means that the log odds would be:

$$\ln\left(\frac{P(rating = medium)}{P(rating = low)}\right) = \beta_{10} + \beta_{11}x_1 + \cdots + \beta_{1n}x_n$$

$$\ln\left(\frac{P(rating = high)}{P(rating = low)}\right) = \beta_{20} + \beta_{21}x_1 + \cdots + \beta_{2n}x_n.$$

Using cross validation approach mentioned above to find the optimal decay parameter, we first built an all-variables model into our regression model, and then

performed variable selection from p-value evaluation (0.05 as a threshold) using two-tailed z-test.

(a) Red wine dataset: Table 1 showed the result from all-variables model. The output in *Coefficients* section has two parts, corresponding to the two equations below:

$$\ln\left(\frac{P(rating = medium)}{P(rating = low)}\right) = 7.84 - 0.24 \cdot fix. - 4.52 \cdot vol. \ldots + 0.32 \cdot alcohol$$

$$\ln\left(\frac{P(rating = high)}{P(rating = low)}\right) = 5.35 - 0.12 \cdot fix. - 7.78 \cdot vol. \ldots + 1.31 \cdot alcohol.$$

The interpretation of these equations, for example, is as follows: In the first log odds, a one-unit increase in the *alcohol* variable is associated with a 0.32 increase in the relative log odds of being in low quality versus medium quality.

From the confusion matrix, we can derive that the test error is 16.56% in the all-variables model. Furthermore, we found that by examining the sensitivity of each class, the model reaches nearly 96% accuracy for the middle class while it has a really poor ability for the low (~6%) and high (~33%) classes, due to the limited amount of training data available in the two groups, and the criteria defined to tune the parameter. This issue would be discussed in the latter algorithm.

Next, we performed variable selection. The z-test showed we cannot reject that *fixed acidity*, *citric acid*, *residual sugar*, *chlorides*, *free sulfur dioxide*, and *total sulfur dioxide* have no relationship to the red wine quality. Based on the test, we removed the variables and build an alternative model. However, such approach did not provide us with lower error rate in this dataset.

(b) White wine dataset: The test error rate is 24.49%, which is higher than the red wine dataset. In addition, same as the result in the red wine dataset, the errors are mainly due to the poor accuracy in low (2.56%) and high (25.44%) class. The z-test showed that *citric acid* and *total sulfur dioxide* have no relationship

to the white wine quality.

(2) K Nearest Neighbor (KNN)

KNN is a non-parametric supervised learning algorithm that manipulates the training set and classifies the test set based on the distance metrics. Therefore, to avoid the noise due to different ranges between the features, normalization is necessary before using KNN. We rescaled the numerical variables by preProc= c("center", "scale") so that each feature has a mean of zero and a standard deviation of one. knn( ) function from *class* package is employed to build the model.

(a) Red wine dataset: The test error rate in KNN model is 15.6% as the best k=29 is chosen (Fig.4-1). However, similar to what we found in logistic regression, when looking into the sensitivity in each class, we found that the model performed high accuracy in the middle class (98%), but low class is really worse (0%), meaning that the model does not have any ability to predict data in lower quality.

(b) White wine dataset: The test error rate 23.47% as the best k=25 is chosen (Fig.4-2). Compared to the red wine set, the model has a little lower accuracy in identifying middle class but better performance in the high-class group (36.4%). Same as the red wine set, the model cannot predict (0%) any low-class observations in the test set.

(3) Decision Tree Method

Decision tree method is a non-parametric supervised learning algorithm that can be used in both classification and regression problem. Each internal node represents a test on a feature, leaf node represents a class label, and branches represent connections of features that lead to those class labels. In this part, we will first build an unpruned decision tree (set cp = 0), and then tuned the complexity parameter c_p from the penalty function using cross validation approach to construct a pruned tree.

(a) Red wine dataset: The unpruned tree has a test error rate of 18.75%, while the pruned tree provides a lower error rate of 15.94%. However, we can see from the confusion matrix (Table 2-1) that, compared to the pruned tree that has no ability to predict low class (0%), the unpruned tree has much greater performance in addressing such class (44.4%). This is because "accuracy" is used during the parameter tuning process, and in order to retrieve a model with higher accuracy, the algorithm would compromise the performance in the low class since decreasing the accuracy here would not have a significant effect on the overall accuracy. According to the pruned tree (Fig. 5-1-2), *alcohol*, *sulphates*, and *free sulfur dioxide* are the most important features in this dataset, since they are on top of the tree to separate the training data.

(b) White wine dataset: The unpruned tree has a test error rate of 24.8% with a really complicated structure (Fig. 5-2-1), indicating that the tree overfits the training set, which is poorly generalize to new samples [3]. After tree pruning, although the model does not have an evident decrease in error rate (24.7%), it provides a more generalized tree that can be easily interpreted and more sustainable in future application. However, similar to the red wine dataset, the model has worse capability in identifying the low-quality group. From the pruned tree (Fig. 5-2-2), we can conclude that the model treats *alcohol*, *pH*, and *chlorides* as the most important variables among the physicochemical properties.

(4) Random Forest

A random forest is a supervised machine learning model that maps data by fitting many decision trees using different bootstrap samples of the original dataset. The algorithm randomly selects a subset of predictors, depending on number of predictors *mtry* set before the calculation, and consider only these few variables as a candidate

for the split. Compared to decision trees, the method improves the overall accuracy of the model while preventing the problem of overfitting. In a special case, when all variables are considered in the split, it is also called bagging. Using *ranger* package, here we first find the optimal *mtry* by cross validation, and evaluate the model performance by examining the test error rate.

(a) Red wine dataset: Using cross validation, mtry=2 is chosen (Fig.7-1), smaller than the typical tuning method: $mtry = \sqrt{p} = 3.31$. The test error rate in random forest is the lowest among all algorithms we used in this project, which is 14.69%. The model has decent accuracy of 98.5% in the middle class and 32.4% in the high class. However, similar to the results from previous methods, it fails to predict all the data in low-quality group.

(b) White wine dataset: Using cross validation to derive the best parameter *mtry*=2 (Fig.7-2), the model has the lowest test error rate (15.1%) among all machine-learning methods. Different from the red wine set, the random forest has a higher accuracy in identifying low group (10.26%), although it is still not a great performance. This result can be explained by the difference in the amount of data in two datasets. Given similar proportions of low-class data to all data (about 4%), the red wine dataset contains only 63 observations in the low-class group, while the white wine dataset contains 183 observations in the low-class group. Larger data allows algorithms to come up with more accurate prediction.

4. Performance Comparison

(a) The performance of four machine learning algorithms in red wine dataset: random forest > decision tree > KNN > logistic regression.

(b) The performance of four machine learning algorithms in white wine dataset: random forest > KNN > logistic regression > decision tree.

5. Reference

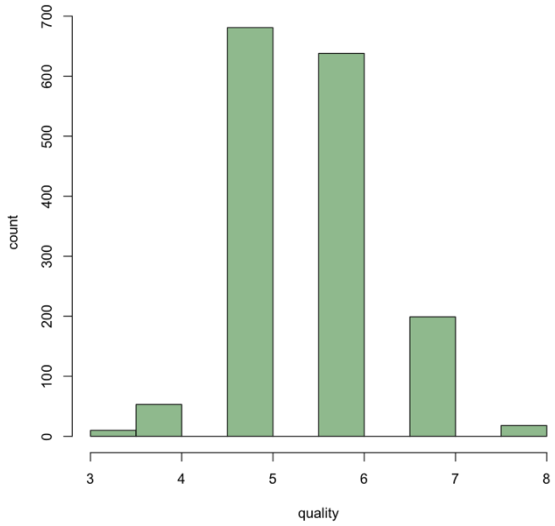
[1] UCI Wine Quality Data Set https://archive.ics.uci.edu/ml/datasets/wine+quality

[2] An Introduction to Statistical Learning

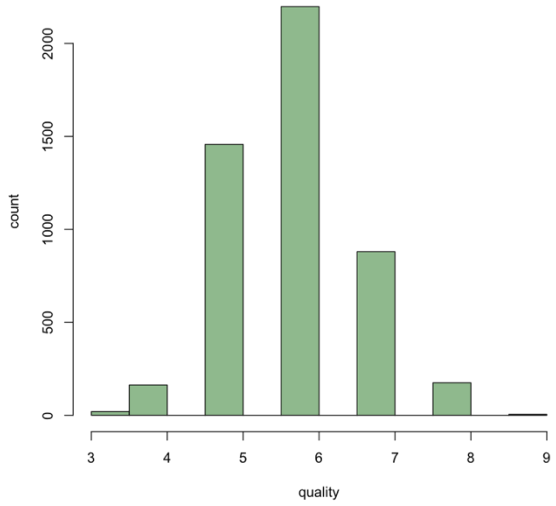
[3] Amro, Asma', Al-Akhras, Mousa, Hindi, Khalil El, Habib, Mohamed and Shawar, Bayan Abu. "Instance Reduction for Avoiding Overfitting in Decision Trees" Journal of Intelligent Systems, vol. 30, no. 1, 2021, pp. 438-459. https://doi.org/10.1515/jisys-2020-0061

6. Appendix

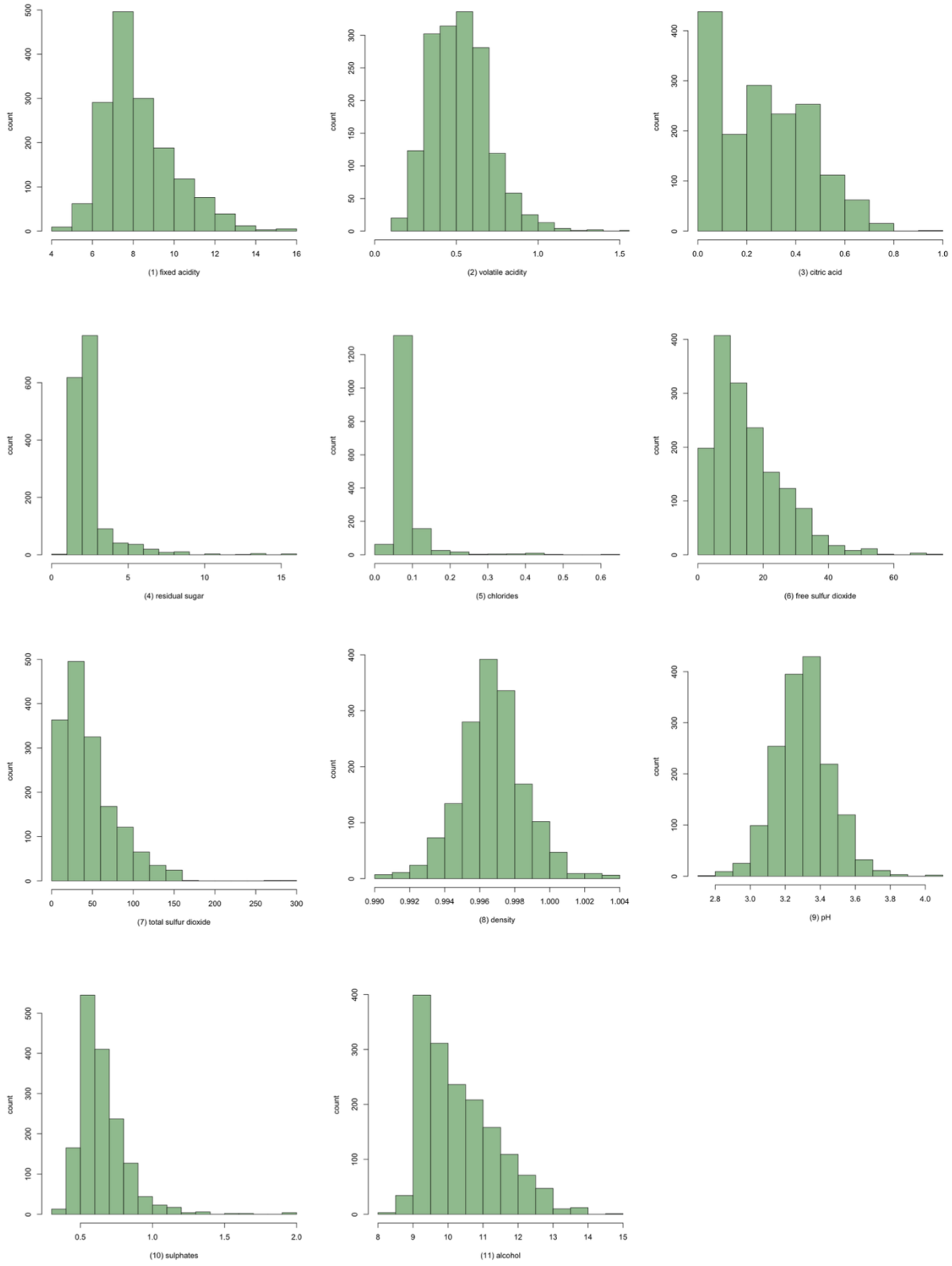
(a) Exploratory Data Analysis



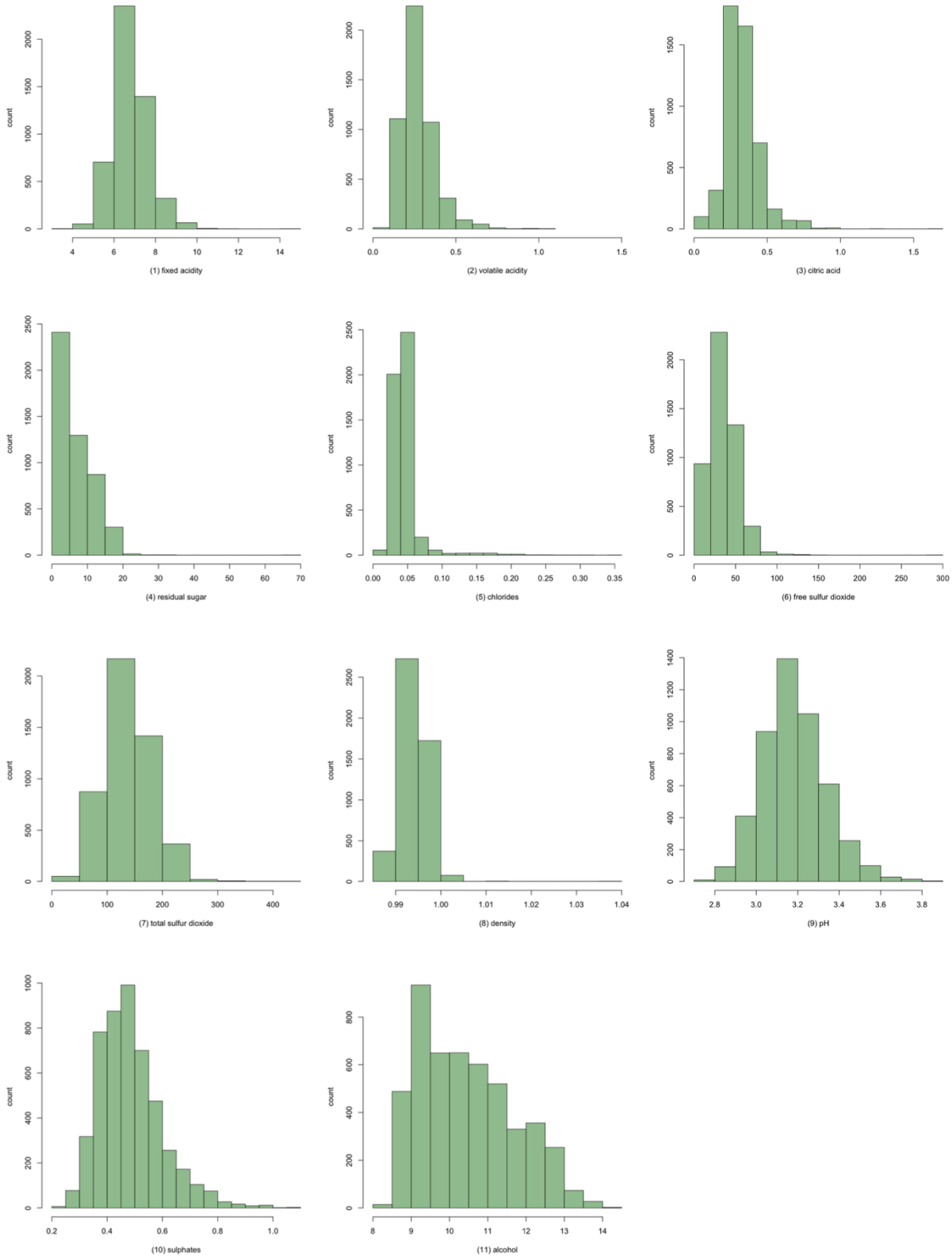
▲ **Figure 1-1. Histogram of quality - red wine**



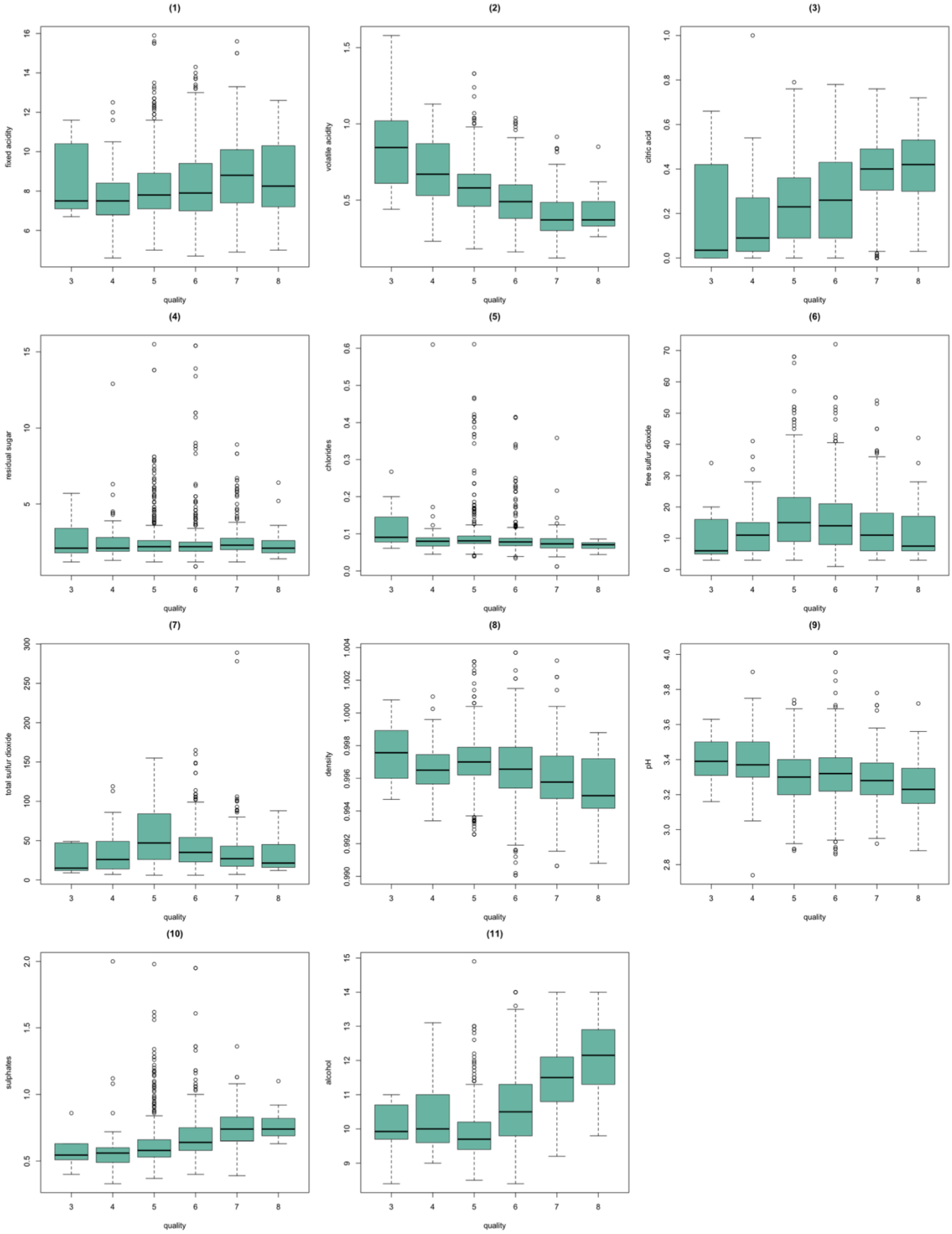
▲ **Figure 1-2. Histogram of quality - white wine**



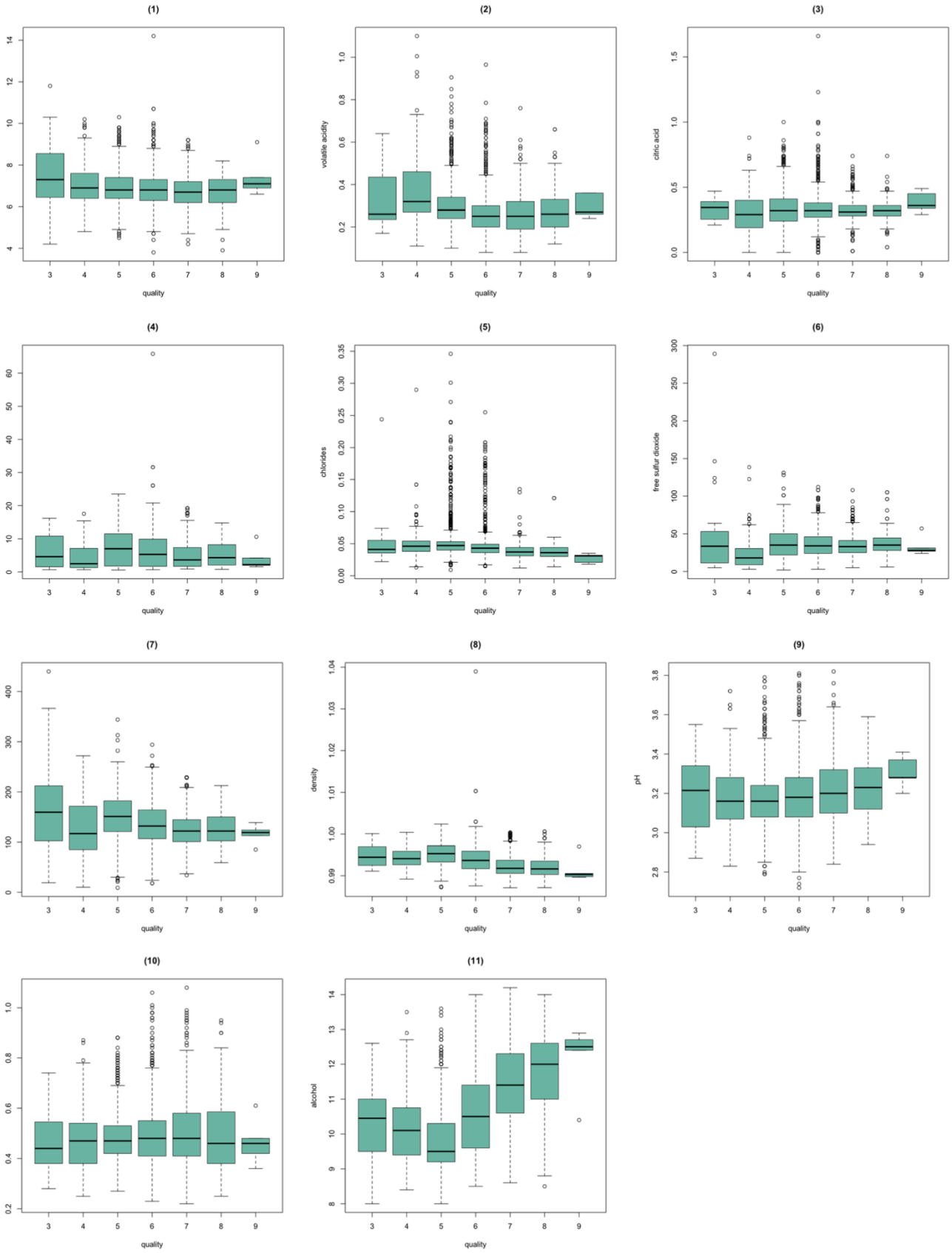
▲ **Figure 2-1-1~2-1-11. Histogram of predictors - red wine**



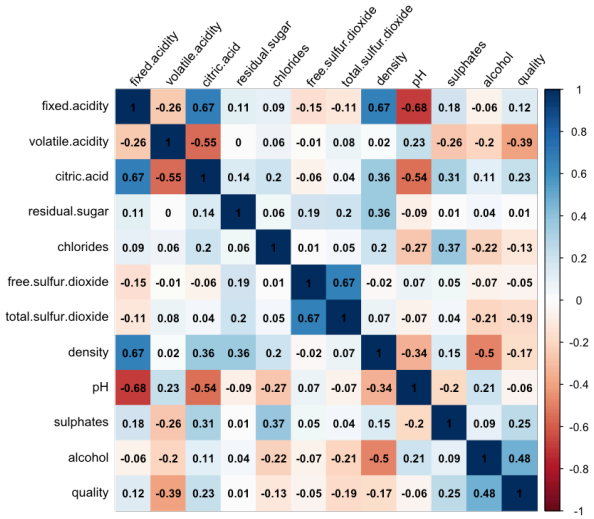
▲ **Figure 2-2-1~2-2-11. Histogram of predictors - white wine**



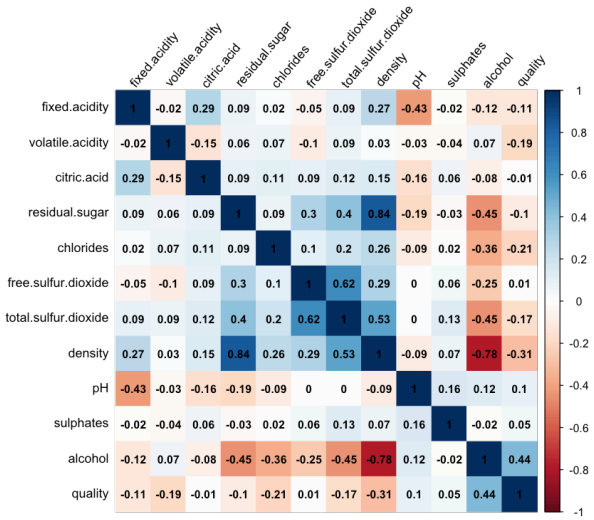
▲ **Figure 3-1-1~3-1-11. Boxplot between predictors and quality – red wine**



▲ **Figure 3-2-1~3-2-11. Boxplot between predictors and quality - white wine**



▲ **Figure 4-1 Correlation matrix – red wine**



▲ **Figure 4-2 Correlation matrix – white wine**

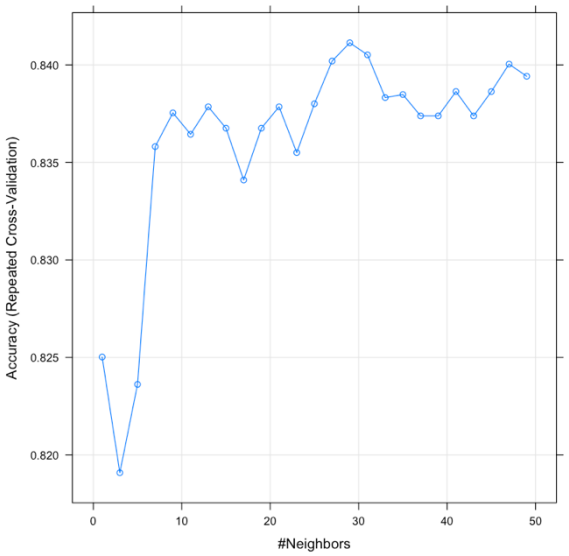
(b) Results

```
Call:
nnet::multinom(formula = .outcome ~ ., data = dat, decay = param$decay,
    trace = FALSE)

Coefficients:
  (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar
1    7.840021    -0.2482966        -4.527252  -0.9517053    -0.07866754
2    5.354792    -0.1279417        -7.786540  -1.0291702     0.04643513
   chlorides free.sulfur.dioxide total.sulfur.dioxide  density        pH
1  -7.520171          0.00602398          0.015641439 11.50890 -4.637648
2 -19.989539          0.01941619          0.001643436  2.79951 -5.275992
  sulphates   alcohol
1  2.022810 0.3205222
2  5.301648 1.3105165

Std. Errors:
  (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar
1    3.192172     0.1796404         1.030319    1.423476      0.1177965
2    3.663522     0.1982999         1.332488    1.671278      0.1340555
  chlorides free.sulfur.dioxide total.sulfur.dioxide  density       pH
1  3.420405          0.02691659          0.009677662 3.117579 1.641361
2  5.608347          0.02992501          0.010950654 3.569242 1.878636
  sulphates   alcohol
1  1.514573 0.2074784
2  1.615854 0.2288743

Residual Deviance: 1028.971
AIC: 1076.971
```

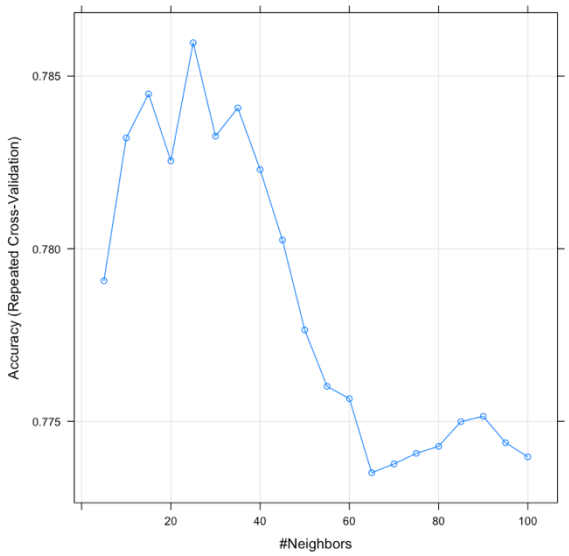
▲ **Table 1-1. Summary from multinomial logistic regression - red wine**

```
Call:
nnet::multinom(formula = .outcome ~ ., data = dat, decay = param$decay,
    trace = FALSE)

Coefficients:
  (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar
1    2.756176    -0.4871695        -4.300555   0.5862385     0.06898397
2   -3.825039    -0.4383921        -7.741417  -0.5086056     0.11539528
  chlorides free.sulfur.dioxide total.sulfur.dioxide   density          pH
1  1.564520          0.03985807         0.0002596997  2.994805 -1.19510705
2 -3.836776          0.05344258        -0.0037451856 -4.167888 -0.05467362
  sulphates   alcohol
1 0.6955212 0.3531185
2 1.9805364 1.1950991

Std. Errors:
  (Intercept) fixed.acidity volatile.acidity citric.acid residual.sugar
1   0.3732450    0.08551564        0.6591896   0.7796042     0.02258332
2   0.3447218    0.09614276        0.8208784   0.8776640     0.02439369
   chlorides free.sulfur.dioxide total.sulfur.dioxide   density        pH
1 0.09171614          0.008326124          0.002868133 0.3674398 0.3419422
2 0.03190236          0.008868888          0.003189214 0.3395349 0.3476191
  sulphates    alcohol
1 0.8780609 0.09139933
2 0.9226163 0.09904228

Residual Deviance: 4355.514
AIC: 4403.514
```

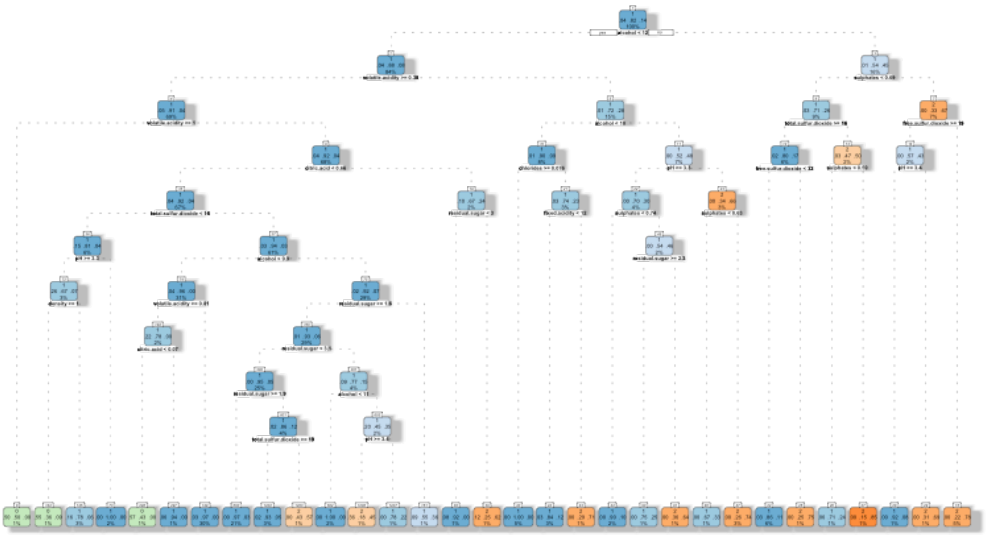
▲ **Table 1-2. Summary from multinomial logistic regression - white wine**



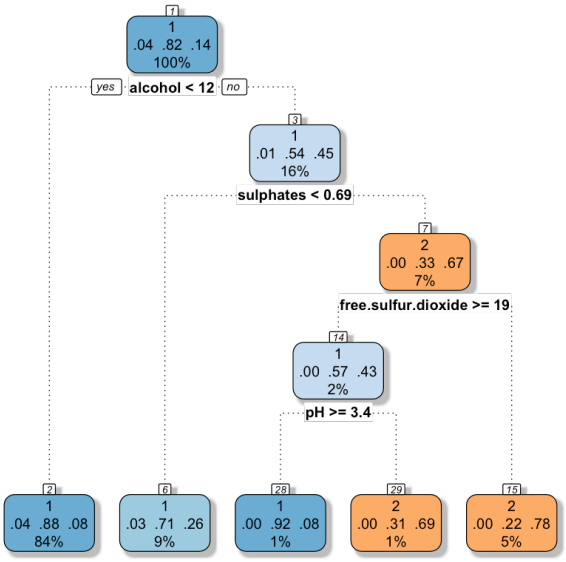
▲ **Figure 5-1. k value selection by 5 repeats 5-fold cross validation - red wine**



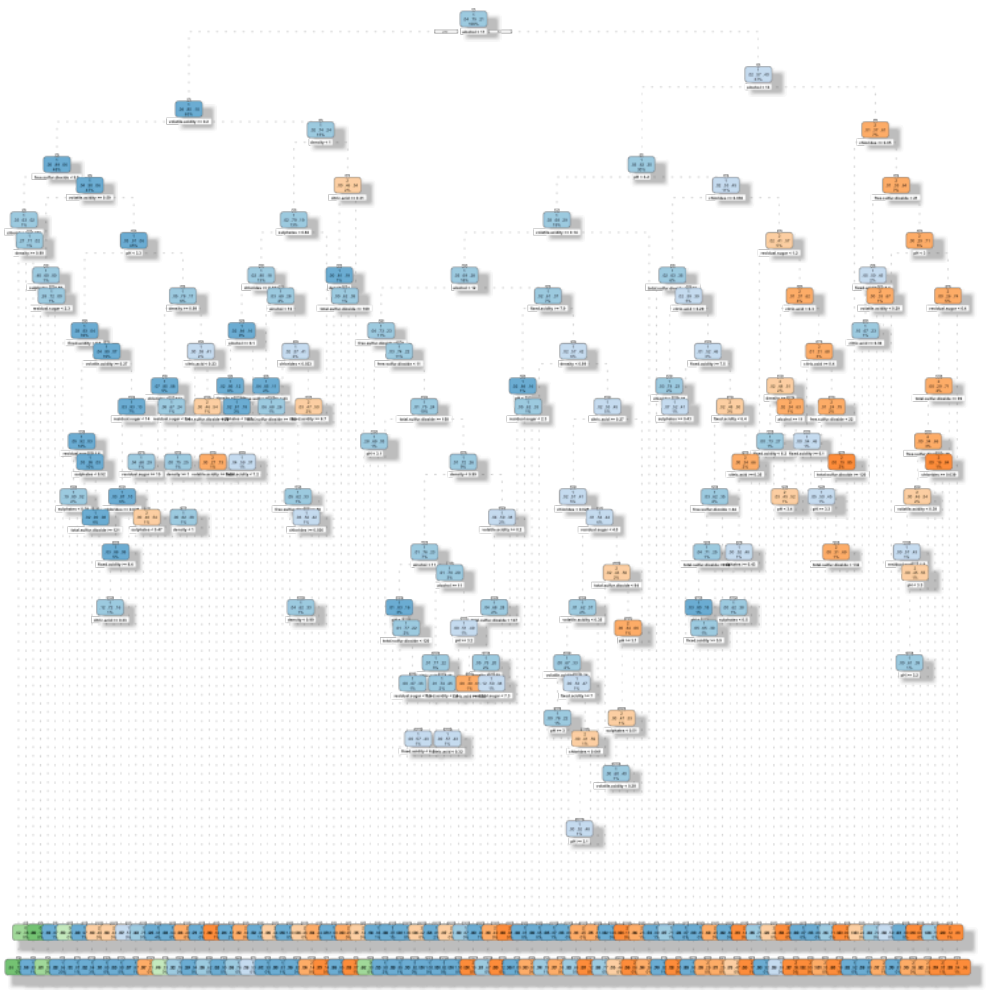
▲ **Figure 5-2. k value selection by 5 repeats 5-fold cross validation - white wine**



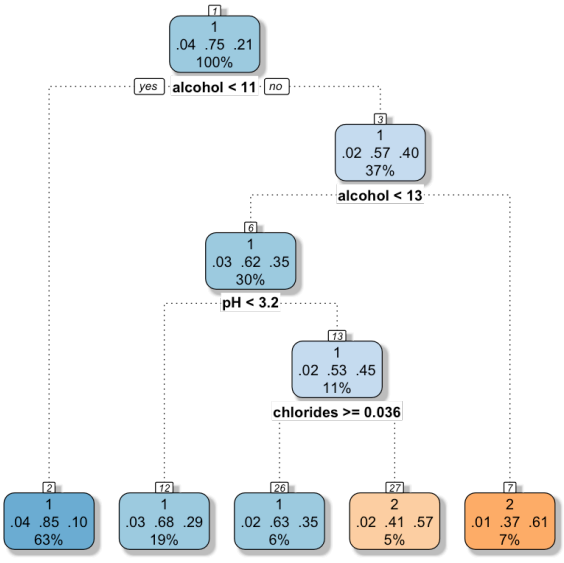
▲ **Figure 6-1-1. unpruned decision tree - red wine**



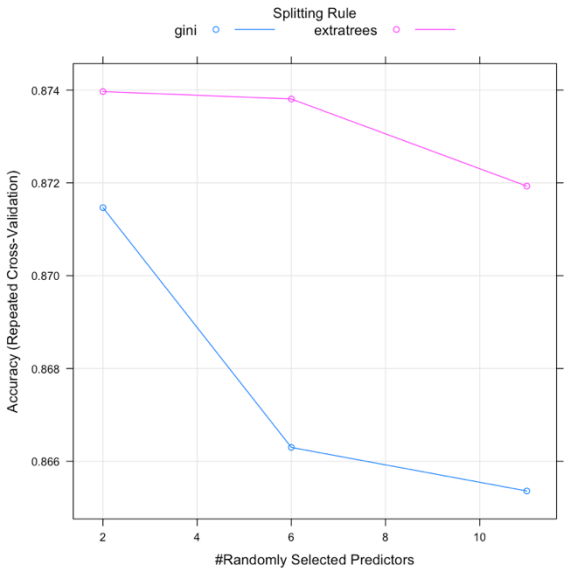
▲ **Figure 6-1-2. pruned decision tree - red wine**



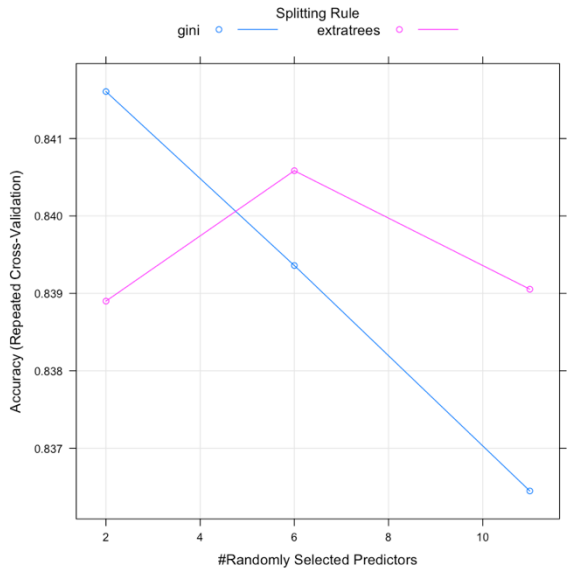
▲ **Figure 6-2-1. unpruned decision tree - white wine**



▲ **Figure 6-2-2. pruned decision tree - white wine**



▲ **Figure 7-1. Tuning *mtry* parameter by 5 repeats of 5-fold cross validation - red wine**



▲ **Figure 7-2. Tuning *mtry* parameter by 5 repeats of 5-fold cross validation - white wine**

(1) Logistic Regression

lr_pred \ rating_test	0	1	2
0	1	2	0
1	17	253	24
2	0	10	13

(3-1) Unpruned Tree

dt_pred1 \ rating_test	0	1	2
0	8	16	0
1	10	236	21
2	0	13	16

(4) Random Forest

rf_pred \ rating_test	0	1	2
0	0	2	0
1	18	261	25
2	0	2	12

(2) KNN

knn_pred \ rating_test	0	1	2
0	0	0	0
1	18	261	28
2	0	4	9

(3-2) Pruned Tree

dt_pred2 \ rating_test	0	1	2
0	0	0	0
1	18	260	28
2	0	5	9

▲ Table 2-1. Confusion matrix - red wine

Test error rate of logistic regression: 16.56%, KNN: 15.63%, unpruned tree: 18.75%

, pruned tree: 15.59%, random forest: 14.69%

(1) Logistic Regression

lr_pred \ rating_test	0	1	2
0	1	0	0
1	37	681	170
2	1	32	58

(3-1) Unpruned Tree

dt_pred1 \ rating_test	0	1	2
0	6	8	1
1	28	614	110
2	5	91	117

(4) Random Forest

rf_pred \ rating_test	0	1	2
0	4	2	0
1	35	679	79
2	0	32	149

(2) KNN

knn_pred \ rating_test	0	1	2
0	0	0	0
1	38	667	145
2	1	46	83

(3-2) Pruned Tree

dt_pred2 \ rating_test	0	1	2
0	0	0	0
1	37	665	155
2	2	48	73

▲ Table 2-2. Confusion matrix - red wine

Test error rate of logistic regression: 24.49%, KNN: 23.47%, unpruned tree: 24.8%

, pruned tree: 24.7%, random forest: 15.1%

z-test-red

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol
0.02176355	3.823791e-06	0.2617375	0.1474355	0.9533217	0.7926574	0.03322567	0	6.942615e-05	0.4210084	9.234175e-03
0.50720581	3.543279e-08	0.4289615	0.6495672	0.2588578	0.7114241	0.99687854	0	2.007141e-03	0.0019257	3.066360e-08

z-test-white

fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol
1.220437e-08	6.84548e-11	0.4520694	2.253284e-03	0	1.692000e-06	0.9278529	4.440892e-16	0.0004739629	0.42829627	0.0001117881
5.120154e-06	0.00000e+00	0.5622524	2.239266e-06	0	1.682241e-09	0.2402635	0.000000e+00	0.8750240078	0.03182098	0.0000000000