Classification of grape cultivars in wine based on chemical characteristics - classification algorithms.

Aleksandra Śledziewska

January 8, 2020

Introduction

The aim of this project is to classify the grape cultivar from which the wine is made on the basis of its chemical characteristics. The chemical characteristics take into account, for instance, color and shade of the wine, alcohol content, alkalinity, magnesium, flavonoid content, etc. Several classification algorithms were applied and the results were compared.

For the analysis, the R program was used, including mainly the **caret** package dedicated to machine learning. In the paper selected classification algorithms were characterized and various functionalities of the **caret** package were presented to facilitate working with data and creating classification models.

The database comes from the UCI Machine Learning Repository, and the original source is: Forina, M. et al, PARVUS - An Extendible Package for Data Exploration, Classification and Correlation. Institute of Pharmaceutical and Food Analysis and Technologies, Via Brigata Salerno, 16147 Genoa, Italy. The data was obtained by chemical analysis of wines from the same region of Italy. It is a balanced database, where each cultivar represents about 1/3 of all observations. The dataset consists of the category variable Cultivar and 13 continuous variables describing the chemical characteristics of the wine. The collection contains 178 observations, i.e. 178 different wines.

The names of all variables contained in the dataset:

- Cultivar - cultivar of the grape from which the wine was produced;
- Alcohol - the amount of alcohol in %;
- Malic acid - malic acid content (one of the main organic acids present in the wine);
- Ash - ash content (inorganic matter remaining after evaporation and burning);
- Alcalinity - alcalinity of ash;
- Magnesium - magnesium content;
- Total phenols - total content of phenols (chemicals influencing taste, color and wine texture);
- Flavonoids - flavonoids content (type of phenols);
- Nonflavanoid phenols - content of phenols other than flavonoids;
- Proanthocyanins - proanthocyanins content (type of phenols);
- Color intensity - intensity of wine color;
- Hue - the shade of wine;
- OD280 - OD280/OD315 of diluted wines (measurements of protein content);
- Proline - content of proline (amino acid present in wines).

```
##    Cultivar Alcohol Malic_acid  Ash Alcalinity Magnesium Total_phenols
## 2         1   13.20       1.78 2.14       11.2       100          2.65
## 3         1   13.16       2.36 2.67       18.6       101          2.80
## 5         1   13.24       2.59 2.87       21.0       118          2.80
## 6         1   14.20       1.76 2.45       15.2       112          3.27
## 8         1   14.06       2.15 2.61       17.6       121          2.60
## 11        1   14.10       2.16 2.30       18.0       105          2.95
## 12        1   14.12       1.48 2.32       16.8        95          2.20
## 13        1   13.75       1.73 2.41       16.0        89          2.60
## 16        1   13.63       1.81 2.70       17.2       112          2.85
## 17        1   14.30       1.92 2.72       20.0       120          2.80
##    Flavanoids Nonflavanoid_phenols Proanthocyanins Color_intensity  Hue
## 2        2.76                 0.26            1.28            4.38 1.05
## 3        3.24                 0.30            2.81            5.68 1.03
## 5        2.69                 0.39            1.82            4.32 1.04
## 6        3.39                 0.34            1.97            6.75 1.05
## 8        2.51                 0.31            1.25            5.05 1.06
## 11       3.32                 0.22            2.38            5.75 1.25
## 12       2.43                 0.26            1.57            5.00 1.17
## 13       2.76                 0.29            1.81            5.60 1.15
## 16       2.91                 0.30            1.46            7.30 1.28
## 17       3.14                 0.33            1.97            6.20 1.07
##    OD280 Proline
## 2   3.40    1050
## 3   3.17    1185
## 5   2.93     735
## 6   2.85    1450
## 8   3.58    1295
## 11  3.17    1510
## 12  2.82    1280
## 13  2.90    1320
## 16  2.88    1310
## 17  2.65    1280
```

First of all, the dataset was examined and it was split into training and test set. Then, all numeric variables were normalized to fit in the range from 0 to 1 for the purpose of future use of algorithms taking into account distance metrics, and basic visualizations were created to better understand the data and correlations between variables. Finally, the classification models were built using sample classification algorithms, such as KNN, LDA, QDA and Random Forest. The results were assessed based on accuracy calculated on the test set.

Methods

Firstly, basic descriptive statistics (mean, standard deviation, median) were calculated for continuous variables in the dataset:

```
## # A tibble: 13 x 4
##    variable             mean     sd       p50
##    <chr>                <chr>    <chr>    <chr>
##  1 Magnesium            " 98.91" " 13.12" 98
##  2 Proline              749.27   319.38   660
##  3 Alcalinity           19.73    3.51     "19.5 "
##  4 Alcohol              12.98    0.77     13.05
##  5 Ash                  " 2.38"  "0.3 "   " 2.36"
##  6 Color_intensity      " 5.02"  2.27     " 4.7 "
##  7 Flavanoids           " 2.02"  1.01     " 2.14"
##  8 Hue                  " 0.95"  0.22     " 0.98"
##  9 Malic_acid           " 2.36"  1.14     " 1.81"
## 10 Nonflavanoid_phenols " 0.36"  0.12     " 0.34"
## 11 OD280                " 2.62"  "0.7 "   " 2.82"
## 12 Proanthocyanins      " 1.57"  0.58     " 1.48"
## 13 Total_phenols        " 2.3 "  0.62     " 2.36"
```

```
## # A tibble: 3 x 2
## # Groups:   Cultivar [3]
##   Cultivar     n
##   <fct>    <int>
## 1 1           41
## 2 2           49
## 3 3           33
```

```
## # A tibble: 3 x 2
## # Groups:   Cultivar [3]
##   Cultivar     n
##   <fct>    <int>
## 1 1           18
## 2 2           22
## 3 3           15
```

The variables take values from very different ranges (from less than 1 to more than 1000). The alcohol content of wine ranges from about 11% to less than 15%. The lowest values are taken by the variable Nonflavanoid Phenols and the highest by Proline. Categorical variable Cultivar consists of 3 categories, with 59 observations belonging to the first category, 71 to the second category, 48 to the third category.

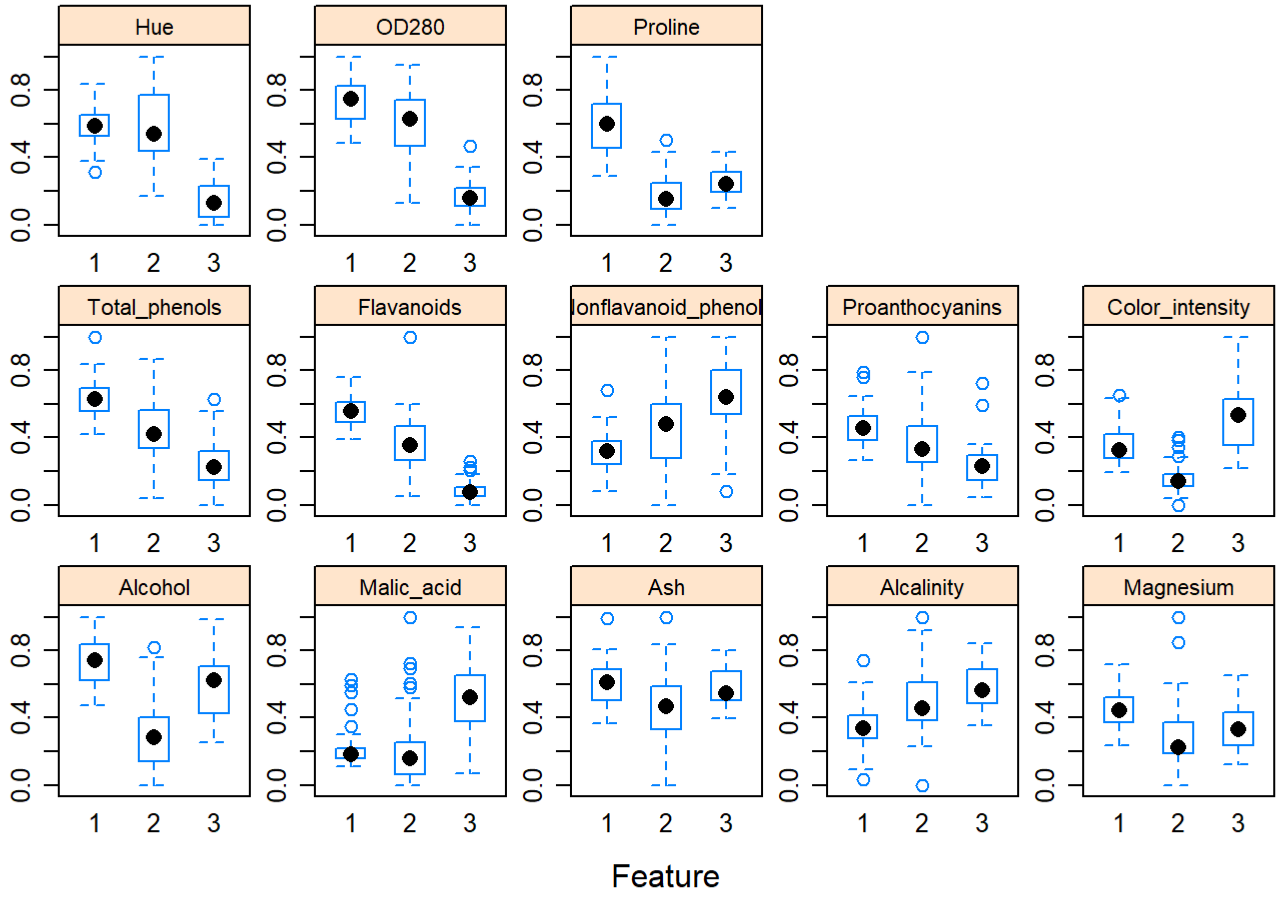
Then, after the initial data exploration, the set was divided into a train and test set. The division was made in a random way in the proportion 70:30, i.e. the train set contains 123 observations and the test set - 55 observations. The **createDataPartition** function from the **caret** package was used to divide the set.

The dataset does not contain null values, i.e. no imputation of missing data needs to be performed, and it does not require transformations to dummy variables (no explanatory categorical variables). However, due to the fact that the variables take values from significantly different ranges, it was necessary to normalize the variables. For this purpose, the **preProcess** function (from the **caret** package) and the **range** method were used, which brings all variables to the range [0,1]. This method is called min-max normalization. The highest value for each variable takes the value of 1 and the lowest one - 0, the rest of the values is scaled to this interval. Normalization begins with the train set, and then, in the same way, using rules known from the train set, the test set is scaled.

```
##     Alcohol Malic_acid Ash Alcalinity Magnesium Total_phenols Flavanoids
## min       0          0   0          0         0             0          0
## max       1          1   1          1         1             1          1
##     Nonflavanoid_phenols Proanthocyanins Color_intensity Hue OD280 Proline
## min                    0               0               0   0     0       0
## max                    1               1               1   1     1       1
```

```
##        Alcohol Malic_acid       Ash Alcalinity Magnesium Total_phenols
## min -0.1472603 -0.0305499 0.1818182 0.04123711 0.1234568     0.1103448
## max  1.1541096  0.9694501 0.7433155 0.76804124 1.1358025     0.9896552
##     Flavanoids Nonflavanoid_phenols Proanthocyanins Color_intensity
## min 0.03375527                 0.02      0.07255521      0.05290102
## max 0.71940928                 1.06      0.90536278      0.81228669
##             Hue       OD280     Proline
## min -0.08045977 0.007843137 0.008559201
## max  1.33333333 1.070588235 0.905135521
```

Afterwards, the next step was a visual representation of the relationships between the Cultivar variable and explanatory variables. This visualization allows for a preliminary assessment of which variables may be most important when deciding on the classification of the wine in a given cultivar. The best variables are those that show significantly different values for different grape cultivars. The **featurePlot** function was used for the visualization, which makes it easy to draw a graph coming from the **lattice** library. Plots were created, such as: boxplot, density plot and scatterplot for variables selected in pairs. Below the most transparent boxplot for all variables is presented.



For example, the variable Hue seems promising in terms of distinguishing cultivars 1 and 2 from cultivar 3, while the variable Proline probably effectively distinguishes cultivars 2 and 3 from cultivar 1. The variable Color intensity, on the other hand, takes the lowest values for cultivar 2, which distinguishes it from cultivars 1 and 3. The least promising variable from this set seems to be the variable Ash, as its values are very similar for each cultivar.

Finally, 4 different classification models were built using:

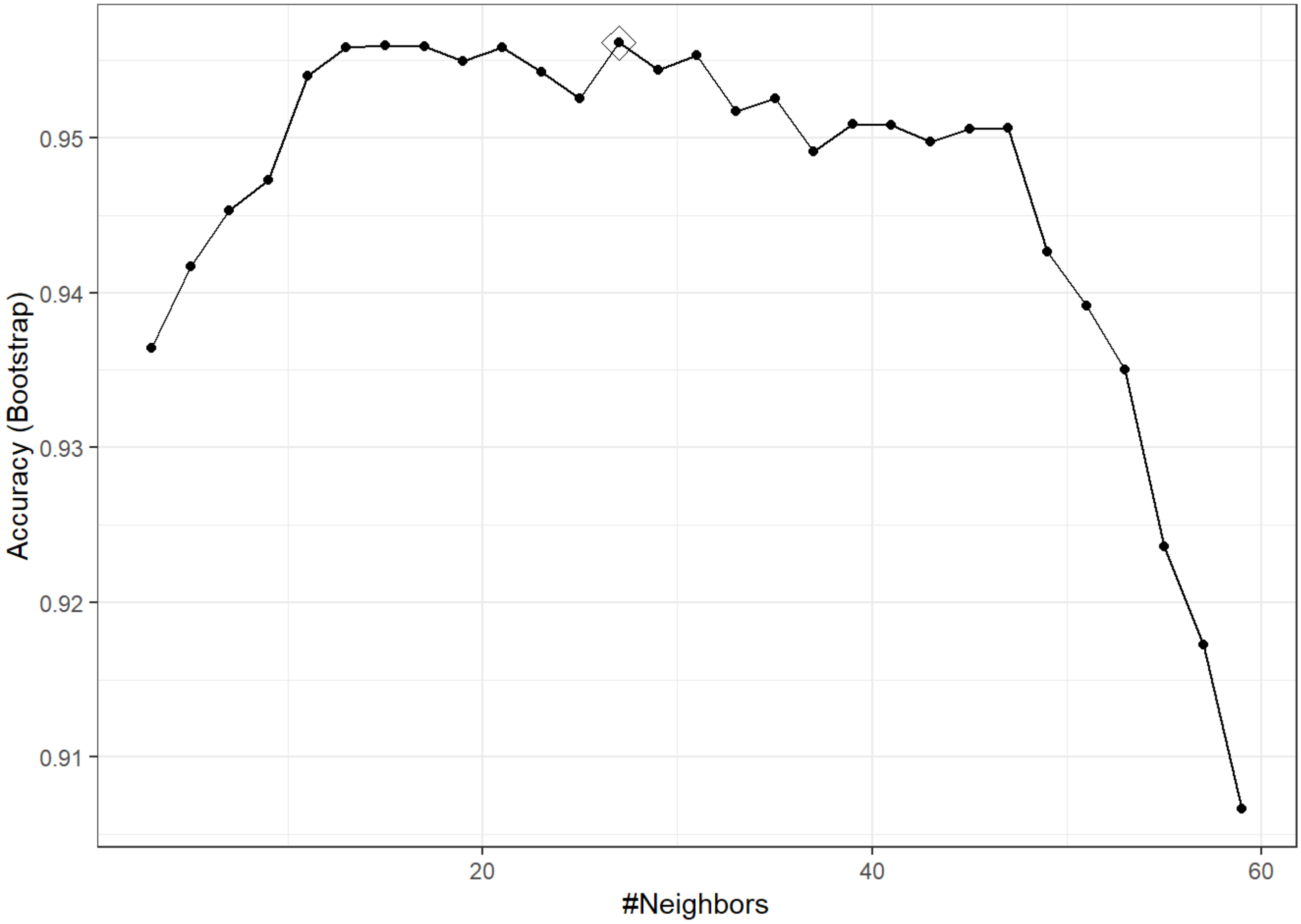
- K-Nearest Neighbors algorithm
- Random Forest algorithm
- Linear Discriminant Analysis (LDA) algorithm
- Quadratic Discriminant Analysis (QDA) algorithm

The **train** function from the **caret** package was used to build the models. This package combines over 200 different algorithms that are scattered across different libraries. **Caret** aims to consolidate all these libraries. The **train** function allows you to train different algorithms using very similar syntax. This function also performs automatic cross validation for different parameters. By default, cross validation is performed by testing on 25 bootstrap samples consisting of 25% randomly selected (sampling with replacement) observations from a given train set. Parameters (depending on the algorithm used) can be changed with **tuneGrid** option in the **train** function.

Results

K-Nearest Neighbors algorithm

The k-nearest neighbors algorithm is a non-parametric method used for classification and regression. This method determines the k nearest neighbors to which the examined observation is closest (usually Euclidean distance is used). A larger number of neighbors leads to smoothing out the division areas, but may lead to larger classification errors. On the other hand, the choice of a very small k-value leads to overfitting.



In the given iteration the number of neighbors k=27 gave the best results in terms of accuracy of matching the model to the data. The k values from the range [3; 59] were considered. The maximum accuracy was equal 96.36%, i.e. the algorithm correctly identifies the class of a given object in over 96% cases.

```
##          Sensitivity Specificity
## Class: 1   1.0000000    0.972973
## Class: 2   0.9090909    1.000000
## Class: 3   1.0000000    0.975000
```

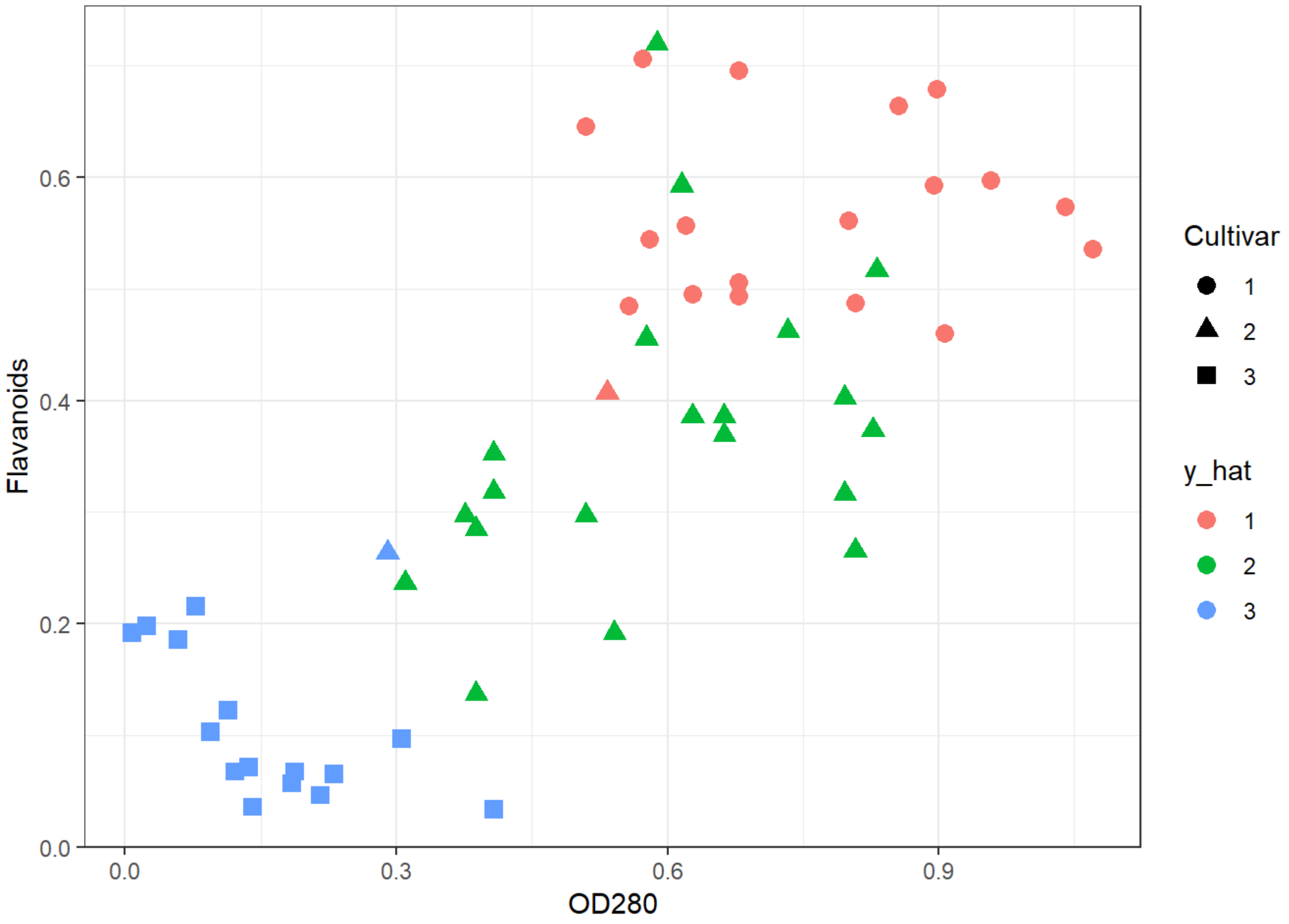
The table shows the sensitivity measures, i.e. the ability of the model to correctly classify observations belonging to a given class, and the specificity measures, i.e. the ability of the model to correctly classify observations not belonging to a given class. The model classifies observations belonging to class 1 and 3 100% correctly, while it rejects observations not belonging to class 2 100% correctly.

```
##           Reference
## Prediction  1  2  3
##          1 18  1  0
##          2  0 20  0
##          3  0  1 15
```

The above measures of accuracy, sensitivity and specificity are based on the contingency table. The table shows that 2 observations were classified incorrectly (both belonging to class 2).

```
## ROC curve variable importance
##
##   variables are sorted by maximum importance across the classes
##                            X1     X2      X3
## Flavanoids           100.000 89.167 100.000
## OD280                100.000 85.000 100.000
## Hue                   98.133 89.271  98.133
## Proline               94.131 94.131  90.539
## Total_phenols         91.908 39.167  91.908
## Color_intensity       78.872 90.000  90.000
## Alcohol               84.405 84.405  48.438
## Proanthocyanins       70.993  7.812  70.993
## Alcalinity            68.877 15.823  68.877
## Nonflavanoid_phenols  60.163  7.083  60.163
## Malic_acid            43.232 44.583  44.583
## Magnesium             40.305 40.305   4.764
## Ash                    8.445  8.445   0.000
```

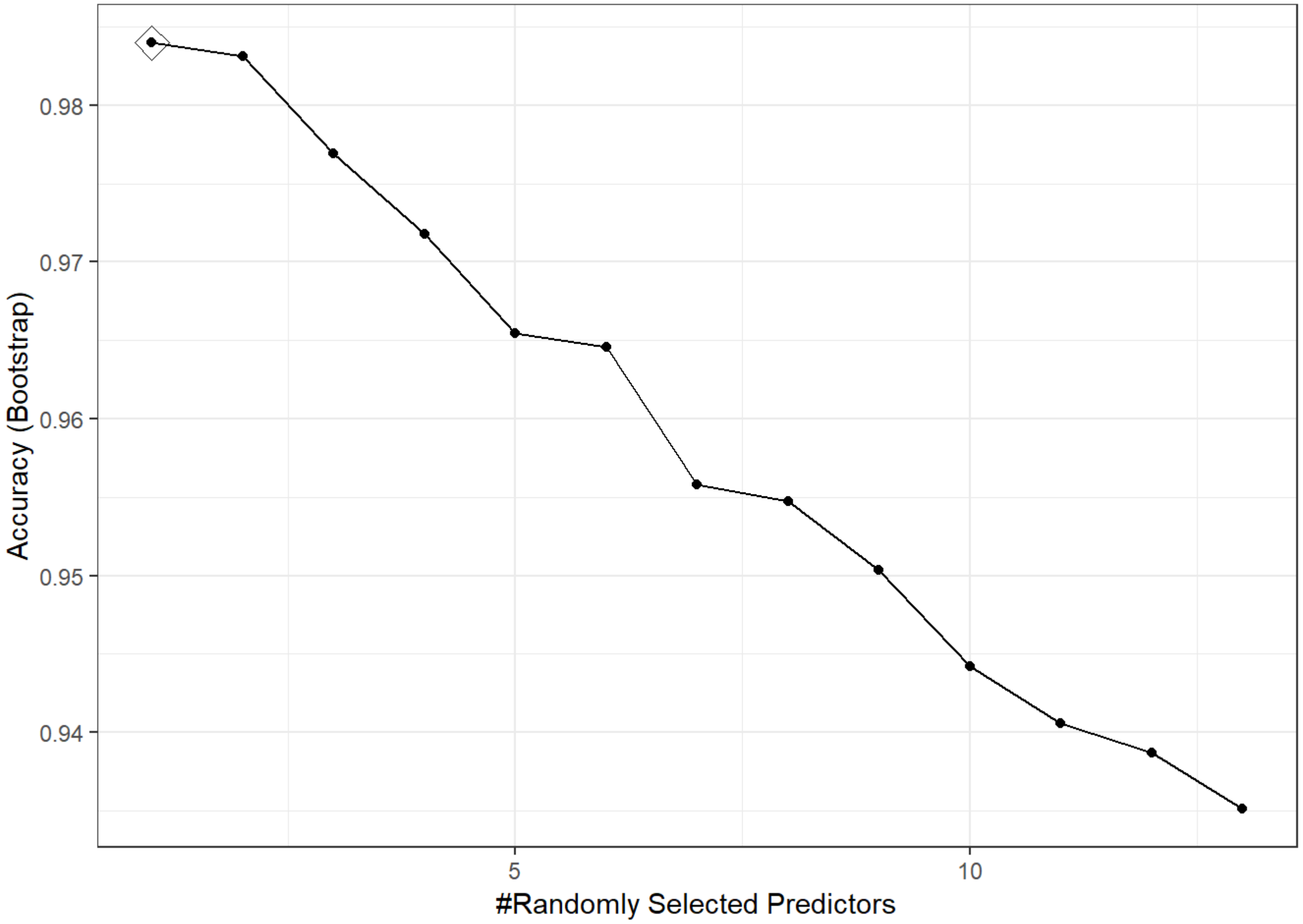
The above table shows a ranking of the variables that best differentiate the grape cultivars in wine, i.e. they are most important for the operation of the algorithm. The three most important variables are: Flavonoids, OD280 and Hue. The weakest variable is the ash content of the wine.



The above figure shows how the class assignment looks in such an algorithm. The graph was created for only two most important variables, so that it can be presented in two-dimensional space. Different colors represent the predicted classes, while shapes represent the classes in reality. Orange triangle and blue triangle represent observations where the algorithm made an error.

Random Forest algorithm

The random forest algorithm is based on the use of decision trees. Each branch coming out of a tree divides it on the basis of the value of a given feature. If a given value (higher or lower than the limit value marked on the branch for continuous features) determines the class of a given observation, the given exit from the decision tree ends with a “leaf”, i.e. an allocation to the class. If a further division by other characteristics is required, another branch of the tree is created. The disadvantage of decision trees is their low stability. Thus a random forest algorithm has been created, which generates a very large number of decision trees and averages their results in order to achieve greater stability of the algorithm.



The number of explanatory variables is a parameter that needs to be determined when building a random forest model. All possibilities were checked - from 1 to 13 explanatory variables. The highest accuracy was achieved with the number of variables equal to 1. The accuracy is then 98.18%, i.e. the algorithm is wrong only in less than 2% of cases.

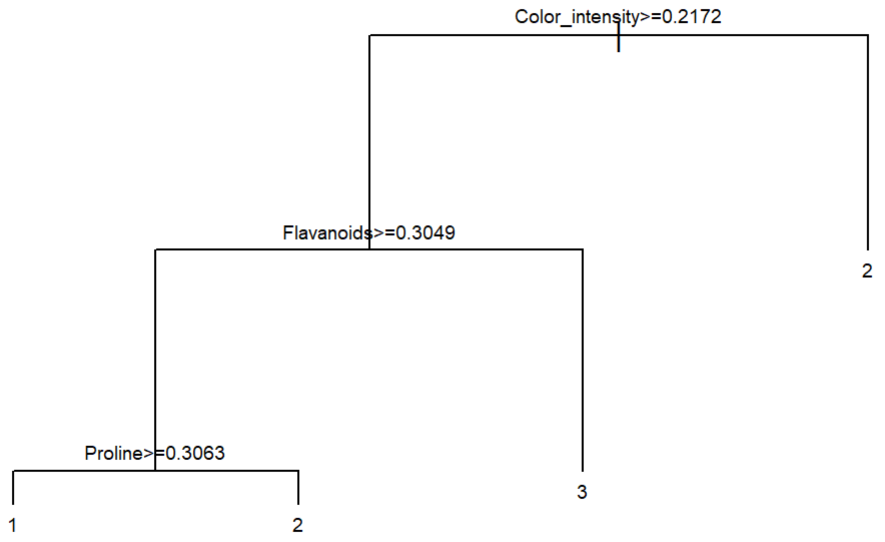
```
##          Sensitivity Specificity
## Class: 1   1.0000000       1.000
## Class: 2   0.9545455       1.000
## Class: 3   1.0000000       0.975
```

```
##           Reference
## Prediction  1  2  3
##          1 18  0  0
##          2  0 21  0
##          3  0  1 15
```

The contingency table and measures of sensitivity and specificity illustrate that the model incorrectly classifies only one observation in the validation set (from class 2). All class 1 observations have been correctly classified.

```
## rf variable importance
## 
##                      Overall
## Proline              100.000
## Color_intensity       96.373
## Flavanoids            68.956
## Alcohol               66.011
## Hue                   59.699
## OD280                 58.549
## Total_phenols         41.459
## Malic_acid            35.905
## Alcalinity            14.161
## Proanthocyanins       12.151
## Magnesium             10.828
## Nonflavanoid_phenols   9.316
## Ash                    0.000
```

The above table shows the variables in order of importance. The most important variable to create a random forest model was the proline content, followed by the intensity of color and flavonoids. When creating a random forest model, most of the variables other than in the k-nearest neighbors model were of the greatest importance. The lowest importance, both for the model of random forest and for the model of k-nearest neighbors, had the variable Ash, i.e. as it could be expected after looking at the boxplots generated in Methodology chapter.



The above graph shows how an example of a decision tree (classification) classifying a grape cultivar in wine looks like. In the first step, the intensity of the color (below 0.22) distinguishes variety 2 from the others. Then the flavonoid content is checked, which, if it is less than 0.30, the variety is classified as class 3. Finally, the proline content is checked. For values less than or equal to 0.31, variety 1 is assigned, and for values greater than 0.31, variety 2 is assigned.

Linear Discriminant Analysis and Quadratic Discriminant Analysis algorithms

The Linear Discriminant Analysis (LDA) algorithm consists in determining the hyperplane separating objects of different classes. The Quadratic Discriminant Analysis (QDA) algorithm is an extension of the linear discriminant algorithm and allows parabolic separating surfaces to occur. However, the quadratic discriminant algorithm works correctly only if the conditional probabilities for explanatory variables have multivariate normal distribution.

```
##           Reference
## Prediction  1  2  3
##          1 18  0  0
##          2  0 22  1
##          3  0  0 14
```

```
##          Sensitivity Specificity
## Class: 1   1.0000000    1.000000
## Class: 2   1.0000000    0.969697
## Class: 3   0.9333333    1.000000
```

The LDA model was only wrong in one case. Its accuracy was equal 98.18%.

```
##           Reference
## Prediction  1  2  3
##          1 18  0  0
##          2  0 22  0
##          3  0  0 15
```

```
##          Sensitivity Specificity
## Class: 1           1           1
## Class: 2           1           1
## Class: 3           1           1
```

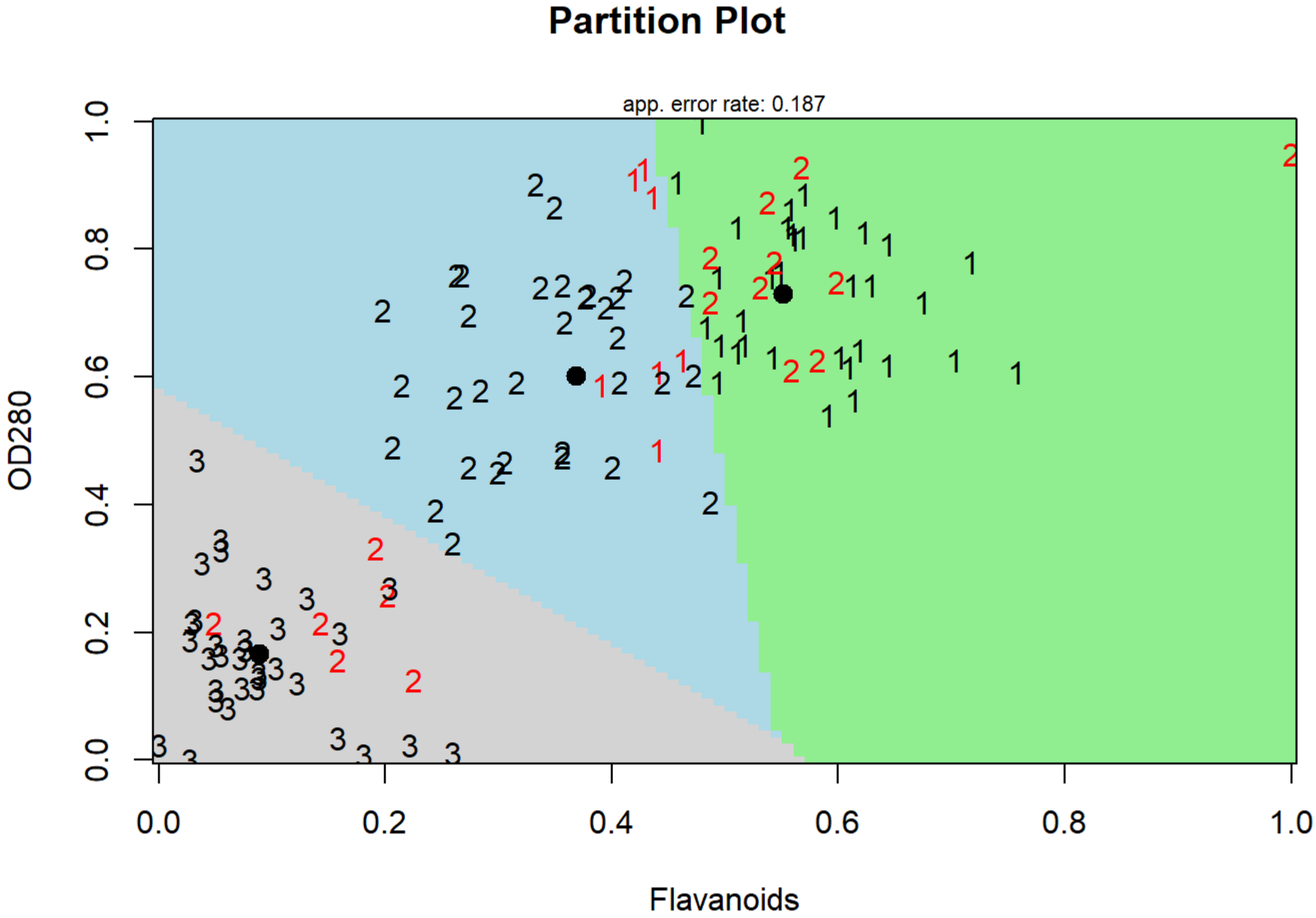
The QDA model was not mistaken even once. The accuracy, sensitivity and specificity of the model were equal 100%.

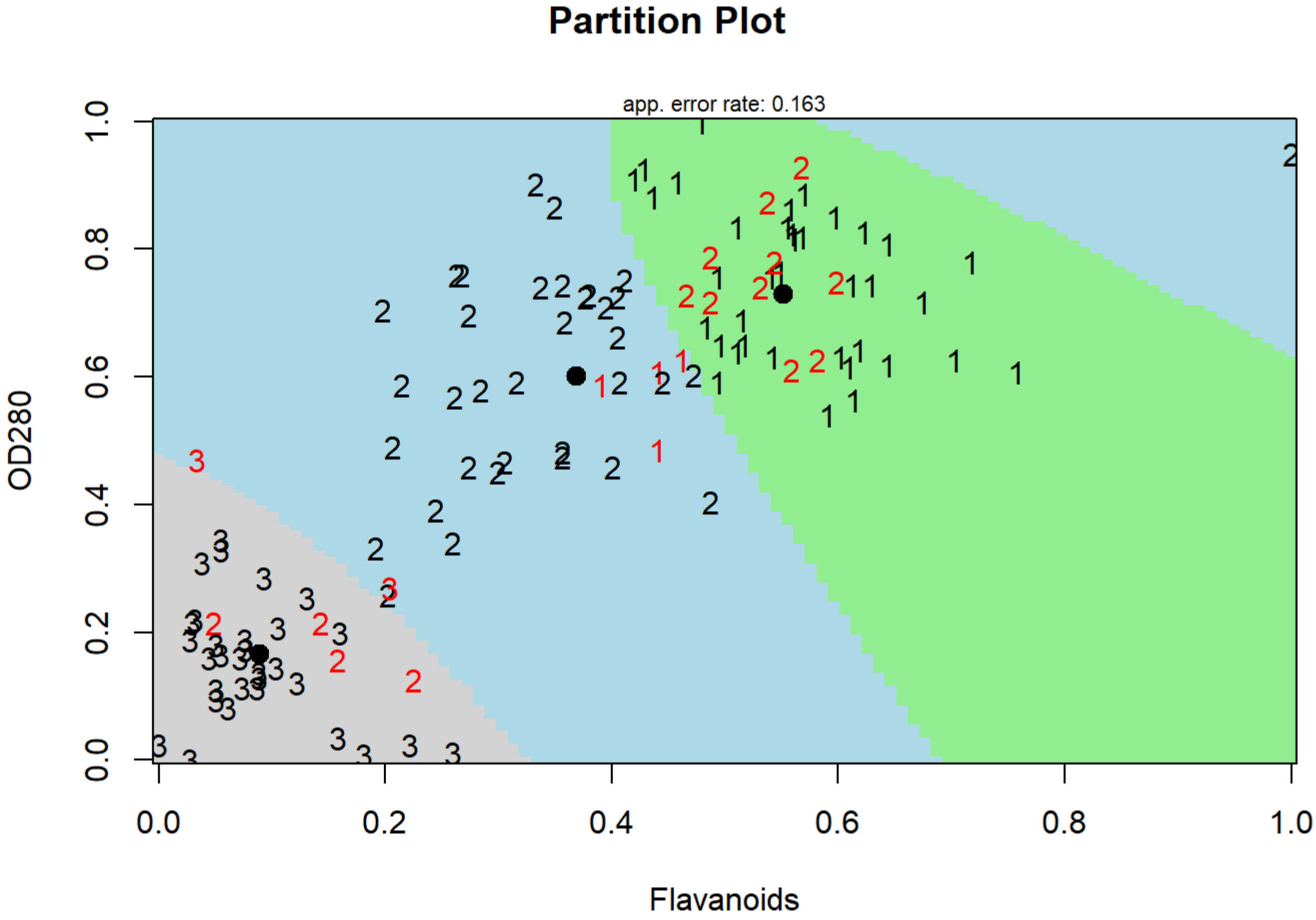
```
## ROC curve variable importance
##
##   variables are sorted by maximum importance across the classes
##                          X1     X2      X3
## Flavanoids           100.000 89.167 100.000
## OD280                100.000 85.000 100.000
## Hue                   98.133 89.271  98.133
## Proline               94.131 94.131  90.539
## Total_phenols         91.908 39.167  91.908
## Color_intensity       78.872 90.000  90.000
## Alcohol               84.405 84.405  48.438
## Proanthocyanins       70.993  7.812  70.993
## Alcalinity            68.877 15.823  68.877
## Nonflavanoid_phenols  60.163  7.083  60.163
## Malic_acid            43.232 44.583  44.583
## Magnesium             40.305 40.305   4.764
## Ash                    8.445  8.445   0.000
```

```
## ROC curve variable importance
##
##   variables are sorted by maximum importance across the classes
##                          X1     X2      X3
## OD280                100.000 85.000 100.000
## Flavanoids           100.000 89.167 100.000
## Hue                   98.133 89.271  98.133
## Proline               94.131 94.131  90.539
## Total_phenols         91.908 39.167  91.908
## Color_intensity       78.872 90.000  90.000
## Alcohol               84.405 84.405  48.438
## Proanthocyanins       70.993  7.812  70.993
## Alcalinity            68.877 15.823  68.877
## Nonflavanoid_phenols  60.163  7.083  60.163
## Malic_acid            43.232 44.583  44.583
## Magnesium             40.305 40.305   4.764
## Ash                    8.445  8.445   0.000
```

In both models the most important variables in terms of ROC curve were: OD280 protein content, flavonoid content and wine hue.

Although the QDA model was not mistaken even once, this can only be a coincidence, because the conditional probabilities of the variables in the model do not seem to have a multivariate normal distribution.


Partition Plot
app. error rate: 0.187
OD280
Flavanoids
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

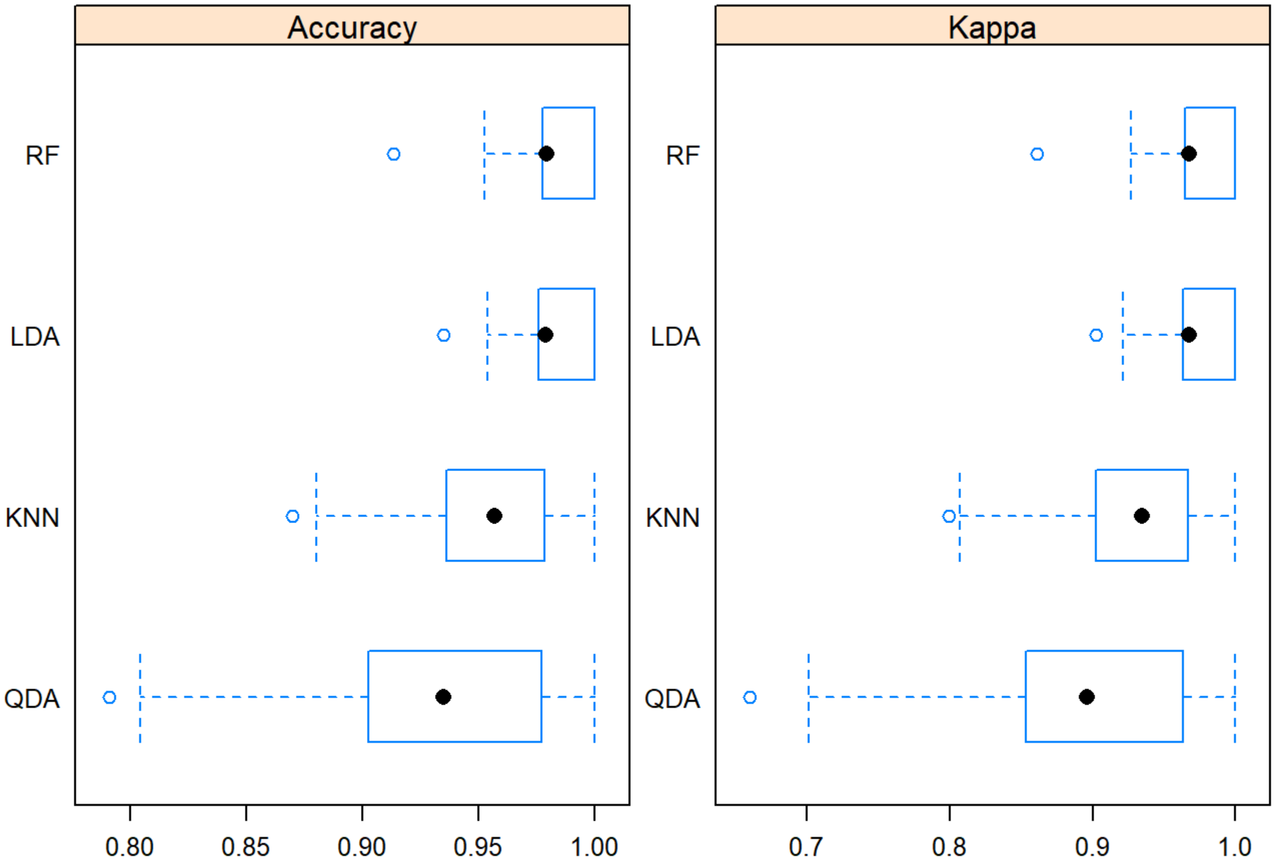

Partition Plot
app. error rate: 0.163
OD280
Flavanoids
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

The above graphs have been created only for the two most important explanatory variables to illustrate the operation of the algorithms on a two-dimensional plane. There are much more misclassified observations (red digits) than for an algorithm with 13 variables. Above diagrams show the differences in the lines marking the boundaries of decision areas. For linear discriminant analysis it is a straight line, and for quadratic discriminant analysis - a parabola. In the diagram for QDA we can also see a significant susceptibility of the algorithm to outliers (digit 2 in the upper right corner). The red digits denote observations for which the algorithm made a mistake and classified them into the wrong group.

Conclusion

In this project the intent was to classify grape cultivars in wine using a dataset with 178 different wines and their 13 chemical characteristics. The data was firstly examined, cleaned and visualized. Then, 4 classification algorithms including the k-nearest neighbors, random forest, linear discriminant analysis and quadratic discriminant analysis were applied to the data. The table and graph below illustrate a comparison of the performance of these algorithms. They compare the accuracy and Kappa coefficient for each model. The Kappa coefficient is very close to the measure of accuracy, but it performs better in unbalanced tests where the classes are of significantly different sizes. In this case we are dealing with a balanced dataset, so more attention can be paid to the measure of accuracy.

```
## 
## Call:
## summary.resamples(object = models_compare)
## 
## Models: RF, KNN, QDA, LDA 
## Number of resamples: 25 
## 
## Accuracy 
##          Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
## RF  0.9130435 0.9772727 0.9791667 0.9839915 1.0000000    1    0
## KNN 0.8695652 0.9361702 0.9565217 0.9561445 0.9782609    1    0
## QDA 0.7906977 0.9024390 0.9347826 0.9283071 0.9767442    1    0
## LDA 0.9347826 0.9756098 0.9787234 0.9803982 1.0000000    1    0
## 
## Kappa 
##          Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
## RF  0.8608169 0.9648118 0.9675895 0.9747726 1.0000000    1    0
## KNN 0.7991266 0.9022869 0.9344729 0.9321311 0.9669194    1    0
## QDA 0.6599297 0.8531782 0.8963893 0.8879984 0.9633188    1    0
## LDA 0.9021970 0.9633274 0.9671329 0.9694916 1.0000000    1    0
```



Despite the initial best result for the quadratic discriminant analysis model (accuracy of 1) in the previous analysis, above table and figure show that this model is subject to the greatest variation in estimation accuracy. In some cases it classifies with 100% accuracy, but it may also be wrong in 20% of cases. This is therefore not the best classification method for such data. This is probably due to the fact that the assumption of multivariate normal distribution of conditional probabilities of the variables has not been met.

Also the k-nearest neighbors algorithm is subject to quite large fluctuations and, on average, performs worse than the random forest and linear discriminant algorithm. The analysis shows that the methods of random forests and linear discriminant analysis are best suited for classifying grape cultivars in wine on the basis of their chemical characteristics. The median for the linear discriminant and random forest model is as high as over 97%, while the minimum accuracy for both models is higher than 90%.

Moreover, the author publishing the dataset mentions that originally this dataset had around 30 variables, not only 13, however, the data was partly lost. More variables would allow to build probably even better model, and it could contain some variables that are strongly different for each cultivar of grapes. However, even with the limited number of explanatory variables, the accuracy of the models is very high.

The model attained in this project is satisfactory, nevertheless, only 4 classification algorithms were tested for this data, while the choice of algorithms is vast. For example, the more modern algorithms based on decision trees could be applied, like AdaBoost or XGBoost implementing gradient boosted decision tree.