Wine Type Prediction With Supervised And Unsupervised Learning

Biljana Simonovikj

06/05/2020

Abstract

This study explores the usage of R programming to predict the wine type with unsupervised learning (PCA and hierarchical agglomerative clustering) and supervised learning (naive_Bayes, LDA and QDA) on benchmark wine database called "Wine Quality" comprised of two datasets for red and white wine of Vinho Verde that were joined together for our analysis. Study data consists of 11 predictor variables as chemical concentrations of physicochemical parameters of laboratoty tests of wine data and a newly created class label called wine type for red and white wine. Prediction of wine type is basically a two-class classification problem. Hierarchical clustering on Principal Components could be applied as an effective technique for its solution. This study considers PCA to be an efficient technique in finding hiden patterns in data, in reducing the dimensionality by removing outliers and redundancy, and in identifing the most correlated variables. Only three Principal Components whose eigenvalues exceeded 1 and whose cumulative proportion of variance was aproximatelly 64.36% of total variance, were sufficiently enough to reconstruct the cluster structure with prominent clusters differentiation of Ward's 2 dendrogram according to the ground truth class labels. Mastering the challenge of imbalanced class labels with post-hoc approach of down sampling, fasciliated mostly supervised classification techniques to be considerd as valuable assets. Metrics for evaluating classification model performances revealed that LDA has the highest accuracy of 0.99 on test data, next is QDA with 0.97 and last is naive_Bayes with 0.96. QDA has the highest sensitivity of 0.99 on train data, while all of the models exhibited prevalence of 0.50 of positive class, "red" type.

Introduction

Data mining is one of the cruical steps in the whole lifecycle of data called Knowledge Discovery in Databases (KDD) process. It refers to a collection of mathematical and computational methods and techniques to extract models and patterns for predictive and descriptive purposes, relying heavily on automated analysis methods that can handle large amounts of data (Dalpiaz (2017)). The most common predictive tasks in data mining, regression and classification, are imperative for decision making. In addition, the most common descriptive tasks, clustering, outlier detection and frequent pattern mining, are efficient in discovering interesting patterns in data (James et al. (2013)).

Vinho Verde is not a grape variety, but it is a product of DOC (The denominação de origem controlada), system of protected designation of origin for wines, as such, it is a unique product in the Minho (northwest) region of Portugal (Johar et al. (n.d.)). The goal of this study is focused on predicting wine types of Vinho Verde into two classes "red" or "white" according to characteristics of variables that are describing physicochemical properties of the wine by supervised and unsupervised data mining approaches while addressing the classification problem with class imbalance of categorical response variable (Cortez et al. (1998)). The first objective of this study is to demonstrate the ability of principal component analysis (PCA) to find the best lower representation of the multivariate data relevant to build the cluster structure by applying hierarchical agglomerative methods of clustering. Concerning model performance accuracy, the second objective is to train supervised generative classification models using train and test wine data with an intention to distinguish fundamental variables with the highest impact on whether a wine type is "red" or "white".

Data

(i) The source of the data is UCI Maschine Learning Repository, https://archive.ics.uci.edu/ml/datasets/Wine+Quality.

(ii) The dataset of interest, called "Wine Quality", is result of experimental study accomplished by pre-processing of protected designation of origin wine samples produced according to Vinho Verde system, collected from May/2004 to February/2007 with a computerized program (iLab) that automatically computes sample testing results, (laboratory and sensory), at official certification entity (CVRVV) in Portugal. The generated dataset contsists of 11 variables (parameters of laboratory tests) without missing values and very unbalanced output variable called "quality", result of sensory test analysis. Originally, it was divided into two datasets for red and white wine, accordingly, that were exploited for classification study conducted by Paolo Cortez in (Cortez et al. (2009)) and donated to UCI MLR.

(iii) The sample size is 1599 rows for red and 4898 rows for white wine.

(iv) Both datasets contain 11 identical input variables and 1 output/response variable. Input variables are continuous/measure and output variable is polytomous, categorical (ordinal) with 6 levels (3, 4, 5, 6, 7 and 8) in red wine dataset and 7 levels (3, 4, 5, 6, 7, 8 and 9) in white wine dataset (Table 1).

Table 1: Data dictionary for red and white wine datasets

Input	Name	Type	Class	Description
1	fixed.acidity	continuous	measure	acidity concentration, (tartaric acid) g/dm3
2	volatile.acidity	continuous	measure	volatile acidity concentraion, g/dm3
3	citric.acid	continuous	measure	citric acid concentration, g/dm3
4	residual.sugar	continuous	measure	residual sugar concentration, g/dm3
5	chlorides	continuous	measure	chlorides concentration, g/dm3
6	free.sulfur.dioxide	continuous	measure	free sulfur dioxide concentration, mg/dm3
7	total.sulfur.dioxide	continuous	measure	total sulfur dioxide concentration, mg/dm3
8	density	continuous	measure	density concentration, g/dm3
9	pH	continuous	measure	estimation of pH level
10	sulphates	continuous	measure	sulphates concentration, g/dm3
11	alcohol	continuous	measure	alcohol percentage in absolute units
Output	Name	Type	Class	Description
1	quality	categorical	category	score of wine quality between 0 and 10

(v) Prepearing data for analysis: a) changing column names of the variables (fixed.acidity, volatile.acidity, citric.acid, residual.sugar and free.sulfur.dioxide by replacing symbol dot, (.) with underscore, (_) in both datasets; b) introducing new class label ("wine_type"), categorical (nominal), to each dataset with *rep()* by rows with labels "red" and "white"; c) joining "red" and "wine" datasets with *rbind()* by rows into united dataset called "wine_data"; d) removing "quality" variable from "wine_data" by column number because our analysis is focused only on "wine_type" class label.

(vi) Pre-processing interventions before unsupervised learning: a) balancing class frequencies of "wine_type" class label with *downSample()* from caret package (Kuhn (2008)); b) separating the predictors (variables 1 to 11) from the class label (variable 12), and converting categorial class label variable into factor variable now called "wine_classe" with two class levels ("red" and "white").

(vii) Pre-processing interventions before supervised learning: a) z-score normalization by columns of predictor variables, where each column is rescaled to have zero mean and standard deviation 1; b) test-train random split of the "wine_data" using caret package *createDataPartition()* function into 80% for training and 20% for testing; c) balancing class frequencies of training data only with *downSample()* from caret package; d) visually exploring correlations of training data with *ggcorr()* from GGally package (Schloerke et al. (2011)), checking normality of each variable with *ggqqplot()* from ggpbur package (Kassambara (2017)) and feature exploratory analysis with *featurePlot()* along with plot type arguments: "box", "ellipse" and "density" from caret package; e) predictor variables (free sulphur dioxide and total sulphur dioxide) were excluded from classification analysis because of strong positive

correlations and the rest of the variables were kept.

(viii) Any other interventions: the class frequencies of testing data were left with the state of nature and reflected the imbalance so that honest estimates of future performance of classifiers can be computed.

Methods

Data treatment and computing performance metrics of data mining algorithms were performed using R Studio (R version 3.6.2, 2019).

Principal Component Analysis (PCA): operates in an unsupervised manner as a dimensionality reduction procedure when we want to find a low-dimensional representation of the multivariate data and plot the observations in this low-dimensional space. Each of the n-observations in the dataset lives in p-dimensional space and each variable could be considered as a different p dimension. For p-dimensional datasets where p>3, it could be very difficult to visualize a multi-dimensional hyperspace. Therefore, we wish to develop a smaller number of artificial variables (called principal components) that will account for most of the variance in the observed variables that may be used as criterion variables in subsequent analyses. These new variables are a linear combination of the original p variables. The amount of variance retained by each principal component is measured by the so-called eigenvalue. The first principal component extracted accounts for a maximal amount of total variance in the observed variables. The second principal component will have two important features: a) it accounts for a maximal amount of variance in the dataset that was not accounted for by the first component and it will be correlated with some of the variables that did not display strong correlations with the first component; b) second feature is that it will be uncorrelated with the first component. In general, the main purpose of PCA is to: a) identify hidden patterns in a dataset; b) reduce the dimensionality of the data by removing the noise and redundancy in the data; c) identify correlated variables, d) data visualisation with corresponding 2D or 3D plots (NOTIONS (n.d.)).

PCA was performed with *pcomb()* from stats package, (R Core Team (2019)) that automatically standardizes the predictors data that was useful because they are expressed in different units (from mg to g, Table 1). The rest of the analysis was performed with functions from factoextra package (Kassambara and Mundt (2017)). The eigenvalues and the proportion of variances retained by the principal components were extracted by *get_eigenvalue()*, while PCA results for each variable were extracted from *get_pca_var()*, where variable coordinates, variable contribution and square cosine were analyzed separately. The analysis was supported with correaltion variable plot with components 1 and 2 obtained with *fviz_pca_biplot()* (Fig 1). In addition, the most contributing variables to the principal components and the quality of representation of all variables (square cosine) on factor map were highlighted with correlation plots obtained with *corrplot()* (Fig 2 and Fig 3).

Hirearchical agglomerative clustering, (HAC): is also unsupervised data mining approach that is best explained by describing the algorithm that begins with a number of clusters equal to the number of observations (a singleton) that are merging in an iterative way according to a given dissimilarity measure between clusters, until there is only one cluster left that represents the entire dataset. The grouping process is visualized with resulting tree-like structure (a dendrogram) that is treated as a single object in all subsequent steps. HAC was performed by utlizing *hclust()* from stats package that provides several methods: single linkage, complete linkage, average linkage, and Ward's minimum variance. The single linkage method computes minimal inter-cluster dissimilarity or specifically, computes all pairwise dissimilarities between the observations in one cluster and the observations in another cluster, and records the smallest of these dissimilarities. The complete linkage method finds similar clusters by computing maximal inter-cluster dissimilarity and records the largest of these dissimilarities while the average linkage method computes mean inter-cluster dissimilarities (Kaufman and Rousseeuw (n.d.)). Particularly, it computes all pairwise dissimilarities between the observations in one cluster and the observations in another cluster, and records the average of these dissimilarities. The Ward's error sum of squares hierarchical clustering method was first published by Joe Ward (Ward Jr (1963)). Two algorithms are implementing Ward's clustering method: Ward 1 described by Murtagh Fionn (Murtagh (1985)) and Ward 2 described in (Kaufman and Rousseeuw (1990)). When applied to the same distance matrix, they produce different results. It was shown that Ward 2 algorithm preserves the Ward's criterion from 1963 (Murtagh and Legendre (2014)) where the dissimilarities

are squared before cluster updating. It is also call a Ward's minimum variance method and note that it requires a squared Euclidean distance matrix.

A dataframe comprising of only first three principal components generated in the previous activity as an output of *get_eigenvalue()* was used for computation of distance or dissimilarity matrix by applying *dist()* from stats package, launch to measure "euclidean" that reurns the distances between the rows of a data matrix. After computing the distance matrix, we started with the default methods: single, average, complete and Ward's 2 methods and proceeded to plot the graphs. Coloured dendrogram with prominent clusters that correspond the best to the two classes of the class label, was visualized with coloured class labels and coloured branches with constructs of code from dendextend package (Galili (2015)). Readjusted class frequencies of class label (wine_class) variable were used for plotting the dendrogram with *downSample()* as described in Data section.

Supervised Classification - Generative Models:

The naïve_Bayes algorithm is one of the most efficient supervised learning algorithms for data mining. However, it is consider as a simple classifier and it is called "naïve" because it utilizes Bayes theorem founded on Bayes' Rule, that simplifies the probabilities of the predictor values by strongly assuming that all of the predictors are conditionally independent from each other with regard to the class which means that the effect of a predictor value on a given class is independent of the values of the other predictors. In real world practical applications, this assumption is often violated and it is not realistic because predictors are often closely related (Kuhn and Johnson (2013)). Namely, Bayes' Rule provides a strong tool to answer the question that based on the predictors that we are analyzing what is the probability that the outcome will be class K for example, with this equation:

$$\mathrm{P}(Y = K \mid X) = \frac{P(X|Y = K) * P(Y)}{P(X)}$$

$\mathrm{P}(Y = K \mid X)$ is posterior probability of the class, $\mathrm{P}(X|Y = K)$ is conditional probability, $\mathrm{P}(Y)$ is the prior probability of the outcome, $\mathrm{P}(X)$ is the probability of the predictor values.

Class probabilities are created and the predicted class is the one associated with the largest class probability. To produce the class probability $\mathrm{P}(X|Y = K)$ for the first class, two conditional probability values will be determined for predictors A and B then multiplied together to calculate the overall conditional probability for the class. For probability of the predictor values (X), everything is the same, except the probabilities for predictors A and B would be determined from the entire training set for both classess. If we visualize distribution of predictors and they are overlapping, it's unlikely that they will be independent, (Witten and Frank (2002)). When the correlation between the predictors is very strong, then is almost unlikely to have a new sample. But if we use the assumption of independence this probability will be overestimated. This model accepts that not all of the predictors to be included in predicting probabilites due to the independence assumption.

The naive_Bayes classification was easily performed with *naive_bayes()* from naivebayes package (Majka (2020)) which output are tables with class conditional probablities for each predictor (the mean and standard deviation of the normal distribution for each predictor in each class) and prior probabilities. The classifications was performed on train and test datasets with *predict()* containing argument "class". Classification metrics were calculated with a *model_classification()* created function according the parameters of confusion matrix and error classification rate was estimated in both data subsets.

The Linear Discriminant Algorithm: uses linear combinations of predictors to predict the class of a given observation. First assumption is that the predictor variables (p) are normally distributed, thus the algorithm is very sensitive to outliers and to imbalanced class frequencies. The second assumption is about homoscedasticity that means the classes have identical variances acros predictor variables (for univariate analysis, $p = 1$) or identical covariance matrices (for multivariate analysis, $p > 1$). The final assumption is about the absence of multicollinearity that means if the predictor variables are correlated, the predictive ability will decrease (Mayor (2015)). The purpose of LDA in this analysis is to find the linear combinations

of the original variables (the 11 chemical concentrations) that gives the best possible separation between the groups (wine types: "red" or "white") in our dataset.

The LDA was easily performed with *lda()* from MASS package (Ripley (2019)) which output contains: prior probabilities of groups, group means and coefficients of linear discriminants. The classifications was performed on train and test datasets with *predict()* in the same way as with naive_Bayes algorithm. Coefficients of linear discriminants and group means were analyzed to estimate the importance of predictors.

Quadratic Discriminant Analysis (QDA): is more flexible than LDA because it does not assumes the equality of variance and covariance, therefore, the covariance matrix is different for each class. If the data can be discriminated using a quadratic function, we can use *qda()* instead of *lda()*. QDA is recommended if the training set is very large, so that the variance of the classifier is not a major issue (Irizarry (2019)). It can be computed using *qda()* for MASS package in exactly the same as *lda()*.

Analysis and Results

Principal Component Analysis: the goal of PCA was to find the best low-dimensional representation of the variation in a multivariate wine dataset. A principal component analysis of the "wine_data" extracted eleven components (with eigenvalues of 3.0298686, 2.4938260, 1.5563470, 0.9705521, 0.7198749, 0.6073177, 0.5231588, 0.5015103, 0.3370240, 0.2276958, 0.0328308 respectively). We examined the eigenvalues to determine the number of principal components to be retained. The first seven components explain 90.01% of the total variance, and the rest of the components up to 11 account for only trivial variance of 9.99%. An alternative criterion that can be used to check the obtained results is to retain enough components so the cumulative sum of proportion of variance explained equals to minimal value or cut off point >= 90%. The PCA output confirmed that cumulative sum of proportion of variance explained demonstrates that the cumulative percentage of variance accounted for by components 1, 2, 3, 4, 5, 6 and 7 is 90.01%, as a result, we have to retain all of the first seven components. By applying PCA to the 11-dimensional space of the predictors, we can see that the first three principal components explain 64.36% of total variance in the "wine_data" which is acceptable large percentage. In addition, first three components have eigenvalues greater than 1 that can be used as another cutoff point to retain principal components because an eigenvalue greater than 1 indicates that those components account for more variance than accounted by one of the original variables in standardized data.

The variable correlation plot (Fig 1) demonstrates clearly that first and second principal components separate red from white wine observations very well. First component separates pH, fixed acidity, chlorides, sulphates, violatile acidity, density from residual sugar, total sulphur dioxide, free sulphur dioxide and alcohol. The second component separates alcohol from total sulphur dioxide, free sulphur dioxide, chlorides, fixed acidity, volatile acidity and density. Examination of the correlation plot demonstrating the most contributing variables to the components (fig 2, left) complements and refines this interpretation because the contributions suggest that component 1 essentially contrasts volatile acidity, fixed acidity, pH and chlorides with total sulphur dioxide, free sulphur dioxide and residual sugar and that component 2 essentially contrasts alcohol with density. The quality of representation of all variables (square cosine) on factor map shown on Fig 2 (right) demonstrates that component 1 contributes highly to total sulphur dioxide, free sulphur dioxide and violatile acidity, while component 2 contributes most to density and alcohol. It appears that second principal component represents its unique physical properties like density, while first principal component represents its chemical properties like concentration of total and free sulphur dioxide additives used as antiseptic during fermentation process and its own sugar levels.

In addition, most of the components show high square cosine values with the first three components that means they are positioned close to the circumference of the correlation circle. The closer a variable is to the circle of correlations, the better we can reconstruct these variables from the first three components and *vice versa*. Thus, we decided to retain only the first three components for further cluster analysis.

Cluster Analysis: we use this study to demonstrate the ability of PCA to extract variables relevant to built the cluster structure. As we discussed previously, hierarchical clustering analysis with single, average, complete and Ward's 2 linkage methods were performed using the first three principal components whose eigenvalues exceeded 1 and whose cumulative proportion of variance was aproximatelly 64.36% of total

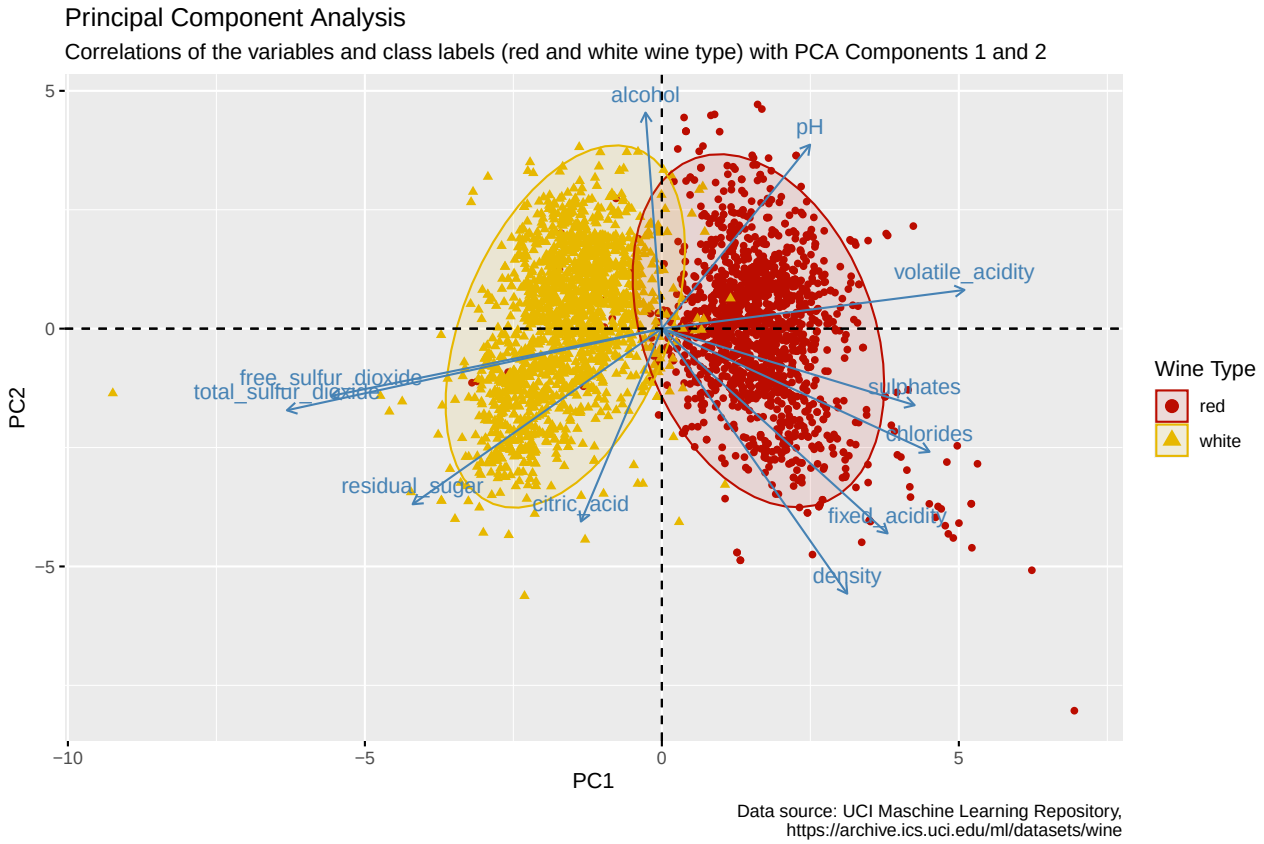

Figure 1: Biplot of individuals (class labels) and variables on Principal Components 1 and 2.

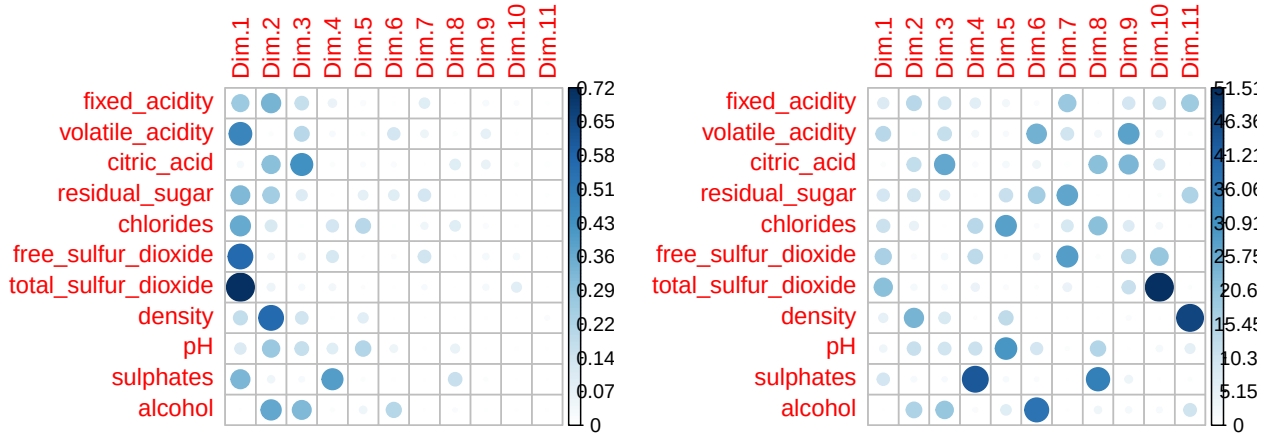

Figure 2: left: The quality of representation of variables on all Principal Components on factor map. right: The most contributing variables for each of the Principal Components.

variance, instead of the original predictor variables. Therefore, the hierarchical clustering performed with that amount the data variance resulted into four dendrograms as compact visualizations of final dissimilarity matrices measured with four different agglomerative methods of hierarchical clustering. The clustering solution that depends on Ward's 2 linkage method outperformed other hierarchical clustering methods. The resulted Ward's 2 dendrogram suggests to group the observations into two prominent clusters with number of observations within the clusters perfectly equivalent to the number of observations that belong to each of the two classes known, as class labels ("red" and "white") according to the ground truth except for few observations being misclassified. It contains two clusters as two branches that occurred at about the same horizontal distance (Brentari, Dancelli, and Manisera (2016)). The outliers are fused rather arbitrarily at higher distance.

In general, Ward's linkage method assumes that agglomeration between two clusters is such that within class-variance of the partition obtained is minimal. The initial cluster distance between observations is defined as squared Euclidean distance. This method measures how much the sum of squares of variance will increase when the pair will merge. With hierarchical clustering, the sum of squares starts out at zero (because each observation is in its own cluster) and then grows as clusters are merging. Ward's method keeps this growth as small as possible. As such, Ward's minimum variance method is very related to PCA because as we know, PCA reduced the dimensionality of "wine_data" into three principal components and amount of variance retained by each component is expressed with eigenvalues. Since both methods are trying to maintain minimal variance, Ward's algorithm revealed the two types of wine as clusters in a completely unsupervised way from PCA loading vectors of only three components.

Supervised Classification Analysis: LDA is sensitive to near zero variance and collinear predictors, thus, we have excluded free sulfur dioxide and total sulfur dioxide from classification analysis because of strong positive correlation coefficient of 0.8 as confirmed with correlation plots that might affect accuracy of LDA and naive_Bayes classifiers. Inspection of box plots and assessment of normality with Q-Q plots demonstrated that data do not appear to be approximately normal, with significant number of highest value observations deviating from a straight diagonal line, especially significant for chlorides, sulphates, residual sugar, citric acid and violatile acidity. Despite that fact, none of the transformations were applied.

Imbalanced class frequencies were known apriori classification as 1280 observations for "red" wine and 3919 observations for "white" wine in the training data. Instead of having the model deal with the imbalance, we attempted to balance the class frequencies with a method called down sampling the majority class, "white type" by removing observations at random until the dataset is balanced. Readjusting class frequencies resulted into equal number of 1280 observations for "red" and "white" wine. The metrics of classification performances of classifiers are presented in the table 2. As expected, LDA is performing well on both train and test data. It has only two false negatives for white type and 7 false positives for red type on train data. QDA has high accuracy, but it is misclassifying red type with 42 false positives, so it is overfitting here. As well, to take into consideration that LDA creates linear decision boundries, while QDA creates quadratic decision boundries. In summary, as we can notice naive_Bayes did not get the chance to show its strengths and it is the worst performing for this data.

Table 2: Performances metrics of naive_Bayes, LDA and QDA classifiers on train and test wine data #

Parameters	LDA.trn	QDA.trn	NB.trn	LDA.tst	QDA.tst	NB.tst
Accuracy	0.98	0.98	0.96	0.99	0.97	0.96
Error_Rate	0.02	0.02	0.04	0.01	0.03	0.04
F1 score	0.98	0.98	0.95	0.99	0.97	0.95
Precision	0.98	0.97	0.95	0.99	0.96	0.96
Recall	0.98	0.99	0.96	0.99	0.98	0.95
Sensitivity	0.98	0.99	0.96	0.99	0.98	0.95
Specificity	0.98	0.97	0.95	0.99	0.96	0.96

In practice, works very well with large number of predictors, even with very small sample size. Also note that, measured prevalence of positive class, (red") is 0.50 for all the classifiers on both train and test data.

Examining the coefficients of the linear discriminant function (there is only one LD) provided an understanding of the relative importance of predictors. The top 5 predictors based on absolute magnitude of discriminant function coefficient are: chlorides(-4.7544), volatile acidity (-2.2979), sulphates (-0.9652), fixed acidity (0.1516) and residual sugar (0.3767). Here, fixed acidity and volatile acidity are inveresly related. Analysis of mean concentrations of these substances by classes revealed that there is a significantly low difference in concentrations for chlorides of only 0.04 g/dm3 between classes, 0.16 g/dm3 for sulphates and 0.24 g/dm3 for volatile acidity. Significantly higher difference in group means concentrations is noticed for fixed acidity of 1.5 g/dm3 and 3.75 g/dm3 for residual sugar. White wine contains 6.3 g/dm3 residual sugar and 6.9 g/dm3 fixed acidity, while red wine contains 2.6 g/dm3 sugar and 8.3 g/dm3 fixed acidity. In conclusion, white wine is more sweet but less acidic while red wine is less sweet but more acidic.

Conclusion

The wine dataset containing 6497 observations in total and 11 variables describing physicochemical properties of the wine with class label "quality" was created by joining red wine dataset with 1599 observations and white wine datasets with 4898 observations. We have replaced "quality" with new class label called "wine_type", thefore, our newly created wine dataset have 11 chemical concentrations describing wine samples from two different types of wine: "red" or "white". We eliminated the fundamental imbalance issue that plagues model training by balancing the frequencies of class labels on training data. In addition, we used balanced class labels in the same way in cluster analysis, to label the observation of resulted dendrograms. We carried out PCA to investigate whether we can capture most of the variation between wine samples using a smaller number of new variables (principal components), where each of these new variables is a linear combination of all or some of the 11 chemical concentrations. As well, we wanted to demonstrate whether extracted principal componenets as the best low dimensional representation of the multivariate wine dataset are relevant to reconstruct the cluster structure built by agglomerative methods of hierarchical clustering. Generative supervised classification methods (naive_Bayes, LDA and QDA) were applied to predict the wine type based on the concentrations of 11 predictors with purpose to select the best performing classifier according to the accuracy and classification error rate for train and test data.

We have shown that variation between wine samples was captured by three principal components whose cumulative proprtion of variance was aproximatelly 64.36% of total variance that was enough to re-built the cluster structure. Observations belonging to each of the two classes known, as class labels according to the ground truth were perfectly captured as two prominent groups in the Ward's 2 dendrogram. At the initial stages of exploratory data analysis, this type of information can be undeniably treasured when the information about class labels is nonexistent at all (Campello, Moulavi, and Sander (2013)).

Summarizing the supervised classification results, we demonstrated that all of the three classifiers performed succesfully on wine data. Nevertheless, LDA showed its strengths because when we have more observations than the number of predictors, then the covariance matrix is invertible, and the data can be accurately divided by a linear hyperplane, then LDA produces a predictively satisfying model. LDA also provided some understanding of the underlying relationships between predictors and the output variable. While white wine has a higher concentraion of sugar level, red wine has higher concentration of fixed acidity.

References

Brentari, Eugenio, Livia Dancelli, and Marica Manisera. 2016. "Clustering Ranking Data in Market Segmentation: A Case Study on the Italian Mcdonald's Customers' Preferences." *Journal of Applied Statistics* 43 (11). Taylor & Francis: 1959–76.

Campello, Ricardo JGB, Davoud Moulavi, and Jörg Sander. 2013. "Density-Based Clustering Based on Hierarchical Density Estimates." In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 160–72. Springer.

Cortez, Paulo, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. 2009. "Modeling Wine

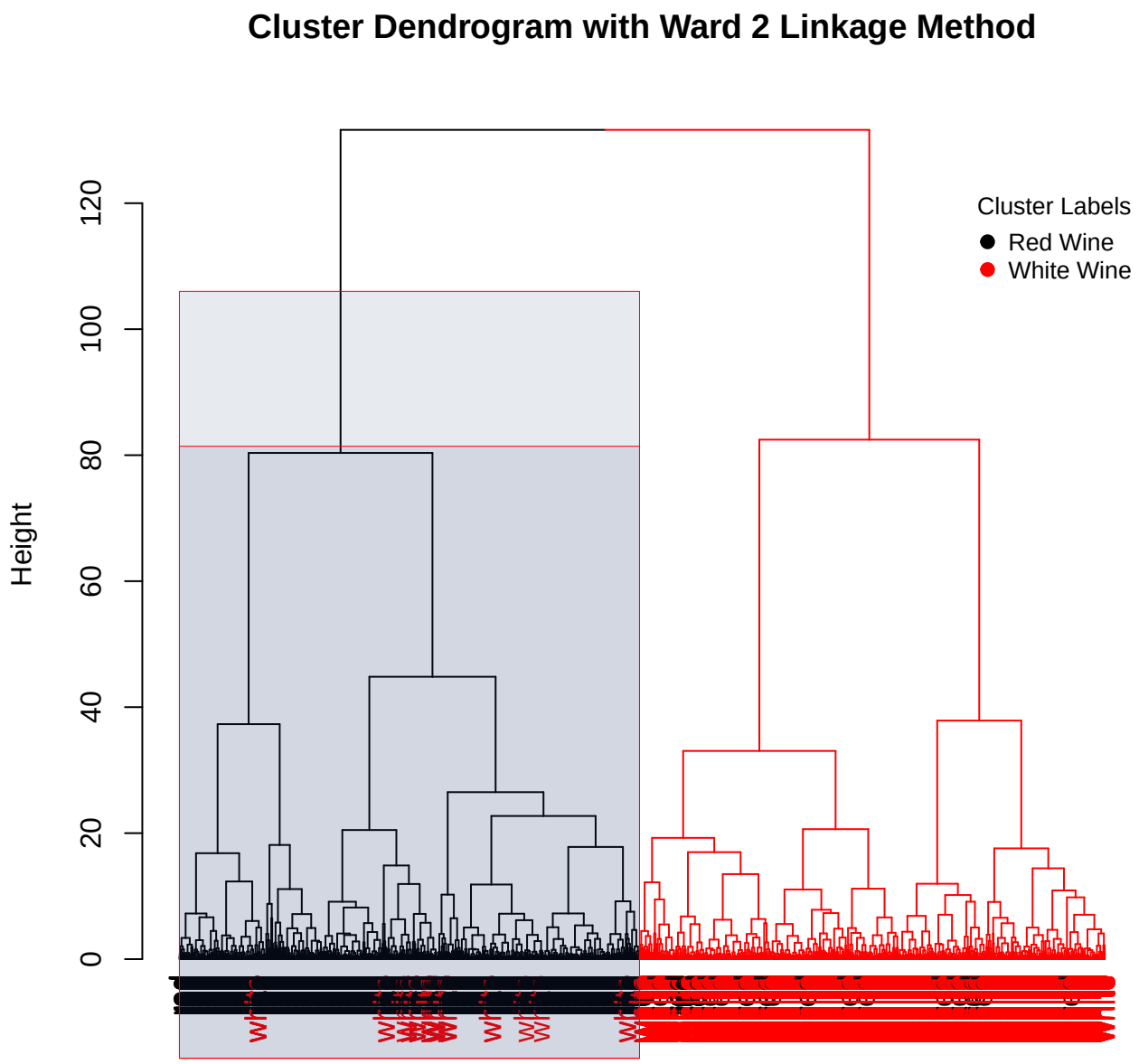

Figure 3: Ward's 2 Dendrogram on scores along PC1-PC3 of wine data 'winequality.csv'

Preferences by Data Mining from Physicochemical Properties." *Decision Support Systems* 47 (4). Elsevier: 547–53.

Cortez, P., A. Cerdeira, F. Almeida, T. Matos, and J. Reis. 1998. "Modeling wine preferences by data mining from physicochemical properties." *Decision Support Systems* 47 (4): 547–53.

Dalpiaz, David. 2017. "R for Statistical Learning."

Galili, Tal. 2015. "Dendextend: An R Package for Visualizing, Adjusting, and Comparing Trees of Hierarchical Clustering." *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btv428.

Irizarry, Rafael A. 2019. *Introduction to Data Science: Data Analysis and Prediction Algorithms with R*. CRC Press.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 112. Springer.

Johar, Sowmya D Sayyed, Shivamogga JNNCE, M Ganavi, and Sankhya N Nayak. n.d. "ANALYZING Wine Types and Quality Using Machine Learning Techniques."

Kassambara, Alboukadel. 2017. "Ggpubr:'Ggplot2' Based Publication Ready Plots. R Package Version 0.1. 6."

Kassambara, Alboukadel, and Fabian Mundt. 2017. "Package 'Factoextra'." *Extract and Visualize the Results of Multivariate Data Analyses* 76.

Kaufman, Leonard, and Peter J Rousseeuw. 1990. "Partitioning Around Medoids (Program Pam)." *Finding Groups in Data: An Introduction to Cluster Analysis* 344. Wiley New York: 68–125.

Kaufman, L Rousseeuw, and P Rousseeuw. n.d. "PJ (1990) Finding Groups in Data: An Introduction to Cluster Analysis." *Hoboken NJ John Wiley & Sons Inc* 725.

Kuhn, Max. 2008. "Building Predictive Models in R Using the Caret Package." *Journal of Statistical Software, Articles* 28 (5): 1–26. https://doi.org/10.18637/jss.v028.i05.

Kuhn, Max, and Kjell Johnson. 2013. *Applied Predictive Modeling*. Vol. 26. Springer.

Majka, Michal. 2020. *Naivebayes: High Performance Implementation of the Naive Bayes Algorithm*. https://CRAN.R-project.org/package=naivebayes.

Mayor, Eric. 2015. *Learning Predictive Analytics with R*. Packt Publishing Ltd.

Murtagh, Fionn. 1985. "Multidimensional Clustering Algorithms." *Compstat Lectures, Vienna: Physika Verlag, 1985*.

Murtagh, Fionn, and Pierre Legendre. 2014. "Ward's Hierarchical Agglomerative Clustering Method: Which Algorithms Implement Ward's Criterion?" *Journal of Classification* 31 (3). Springer: 274–95.

NOTIONS, PREREQUISITE. n.d. "Principal Component Analysis."

R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Ripley, Brian. 2019. *MASS: Support Functions and Datasets for Venables and Ripley's Mass*. https://CRAN.R-project.org/package=MASS.

Schloerke, Barret, Jason Crowley, Di Cook, Heike Hofmann, Hadley Wickham, François Briatte, Moritz Marbach, Edwin Thoen, Amos Elberg, and Joseph Larmarange. 2011. "Ggally: Extension to Ggplot2."

Ward Jr, Joe H. 1963. "Hierarchical Grouping to Optimize an Objective Function." *Journal of the American Statistical Association* 58 (301). Taylor & Francis Group: 236–44.

Witten, Ian H, and Eibe Frank. 2002. "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations." *Acm Sigmod Record* 31 (1). ACM New York, NY, USA: 76–77.