POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Informatica

Data Spaces Tesina

Red Wine Quality

Candidato
Elia Bravo

Ottobre 2019

Indice

Capitolo 1

Introduction

1.1 The Dataset

The goal of this report is to analyse a dataset related to red variant of the Portuguese "Vinho Verde" wine and apply on it some Unsupervised and Supervised Learning methods studied during the Data Spaces course. The wine dataset can be used for classification or regression tasks. We have 11 quantitative variables based on physicochemical tests:

1. fixed acidity: most acids involved with wine or fixed or nonvolatile (do not evaporate readily).

2. volatile acidity: the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste.

3. citric acid: found in small quantities, citric acid can add 'freshness' and flavor to wines.

4. residual sugar: the amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweet. We can consider a division between "sweet" and "no sweet" wines.

5. chlorides: the amount of salt in the wine.

6. free sulfur dioxide: the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine.

7. total sulfur dioxide: amount of free and bound forms of SO2; in low concentrations, SO2 is mostly undetectable in wine, but at free SO2 concentrations over 50 ppm, SO2 becomes evident in the nose and taste of wine.

8. density: the density of water is close to that of water depending on the percent alcohol and sugar content.

9. pH: describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale.

10. sulphates: a wine additive which can contribute to sulfur dioxide gas (S02) levels, wich acts as an antimicrobial and antioxidant. The last variable based on sensory data:

11. quality (score between 0 and 10): this is the only variable that assumes discrete values, so it's the only qualitative variable, that will be used as output variable for classification tasks.

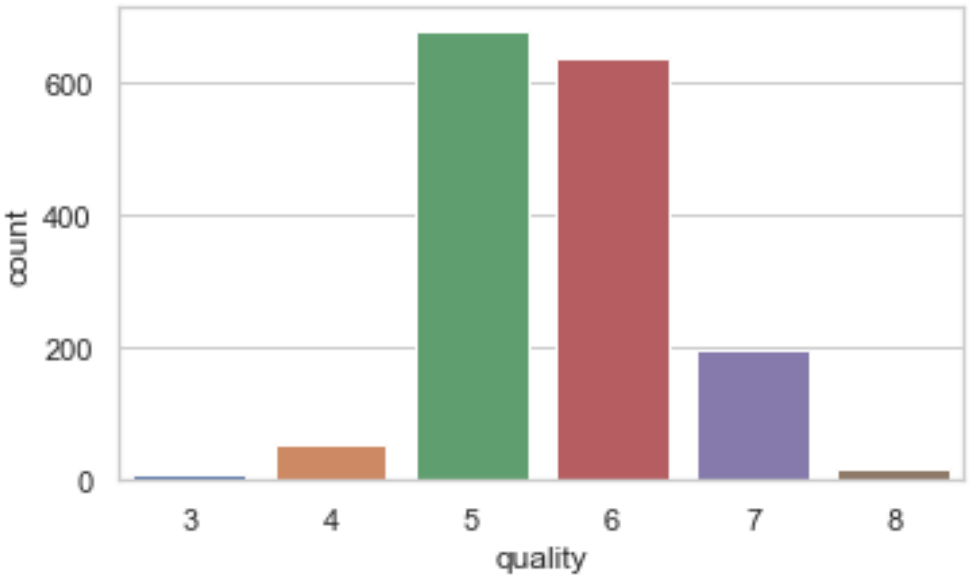

Figure 1.1. quality - count

1.2 First Analysis

The preliminar analysis of the dataset can be done on the quantity of the wines with regard to the different quality values. The first plot shows that we don't have a balanced dataset, but there's a majority of 'normal' wines. Then, the boxplots give us a representation of the distribution of the values of each quantitative variable, in relation with the quality. So we can make a first evaluation of how a certain range of a variable can be related to the quality of the wine, and try to make a hypothesis about some linear proportionalities between a measured feature and the classification of the product.

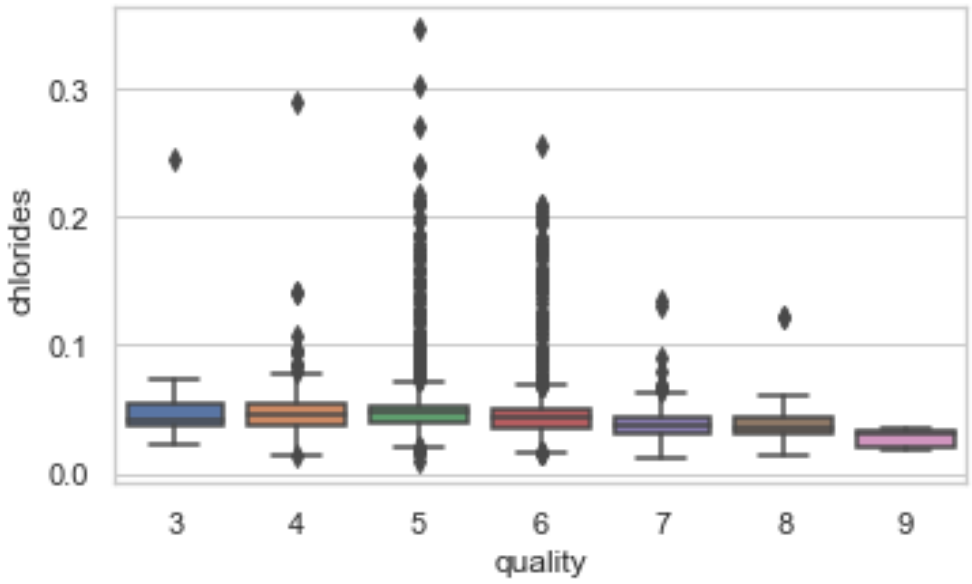

Figure 1.2. boxplot: quality - chlorides — The quality is higher when the quantity of chlorides tends to decrease, and this linearity is more and more evident for very good wines (see the wines rated with '9'). Moreover, in the plot, the outliers with a very high chlorides are classified as bad wines

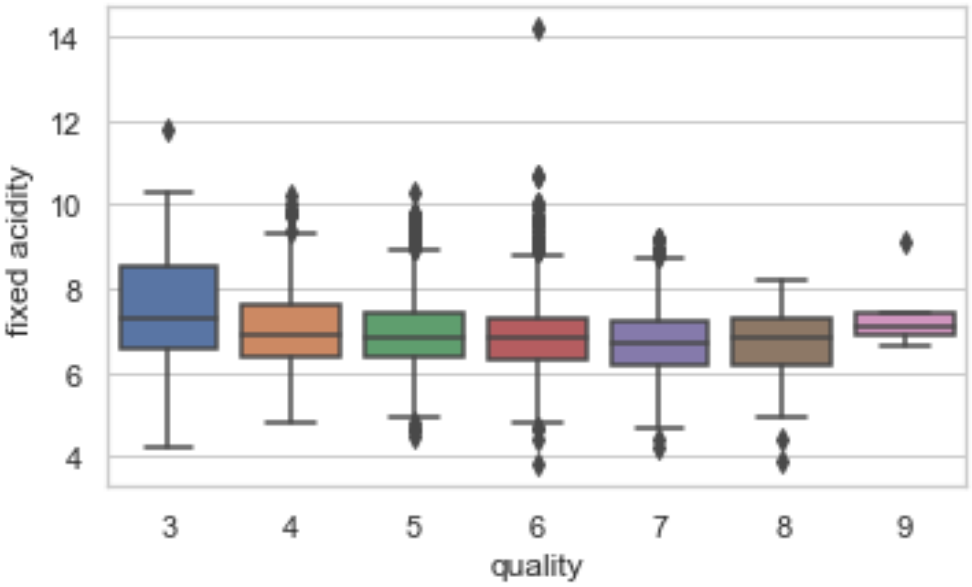

Figure 1.3. boxplot: quality - fixed acidity — A very high value for fixed acidity (around 8) is a not always cause of low quality, as we can see in the outliers of the wines rated with 9. Nevertheless, wines with a acidity that is more than '10' are not considered good wines in our dataset.

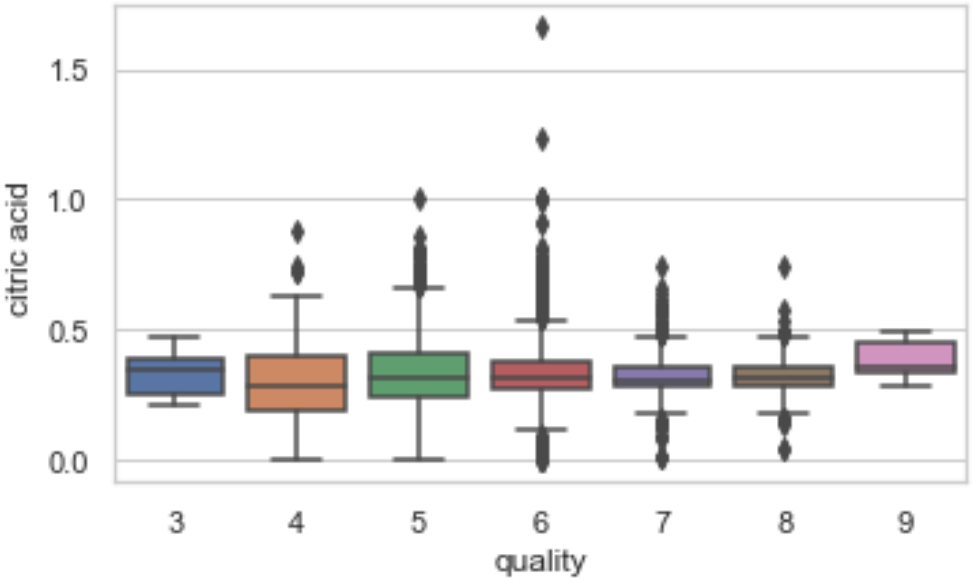

Figure 1.4. boxplot: quality - citric acid — This variable looks not to affect directly the quality of the wine.

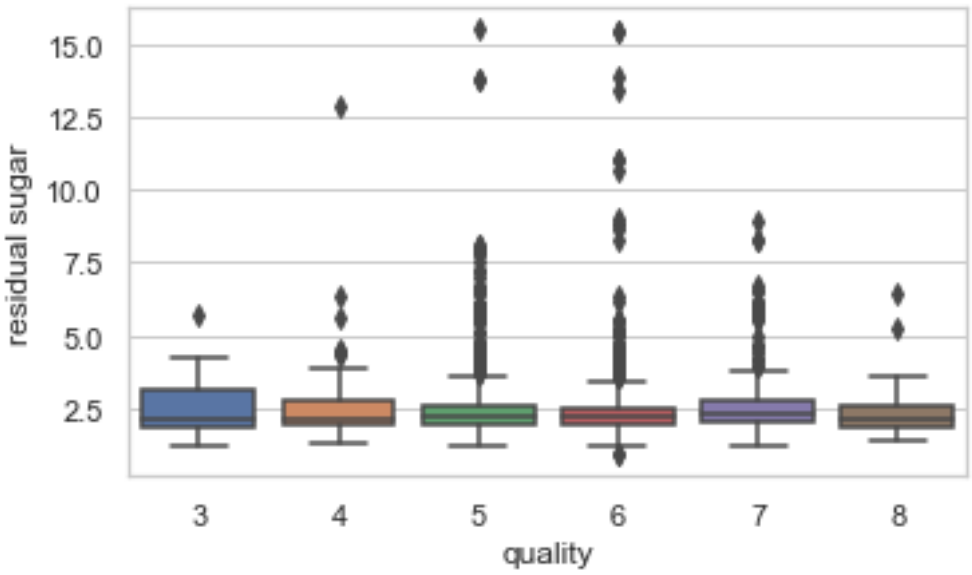

Figure 1.5. boxplot: quality - residual sugar — Even here there are good and bad wines, sweet and less sweet

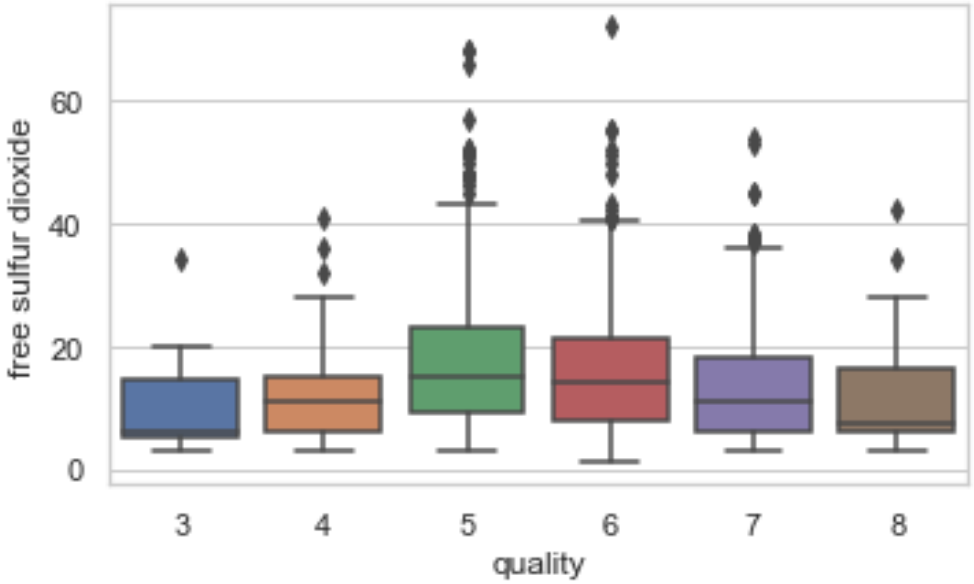

Figure 1.6. boxplot: quality - free sulfur dioxide

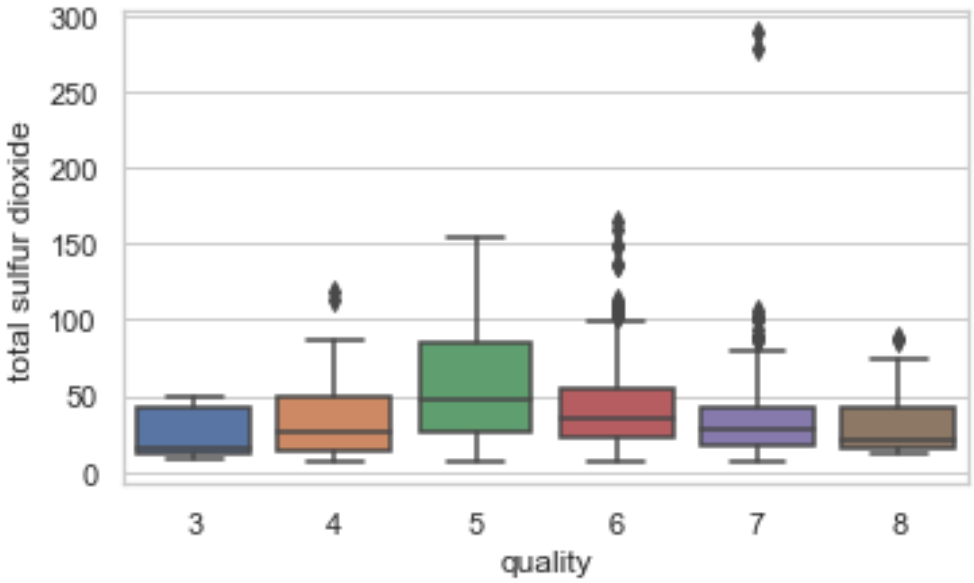

Figure 1.7. boxplot: quality - total suldur dioxide

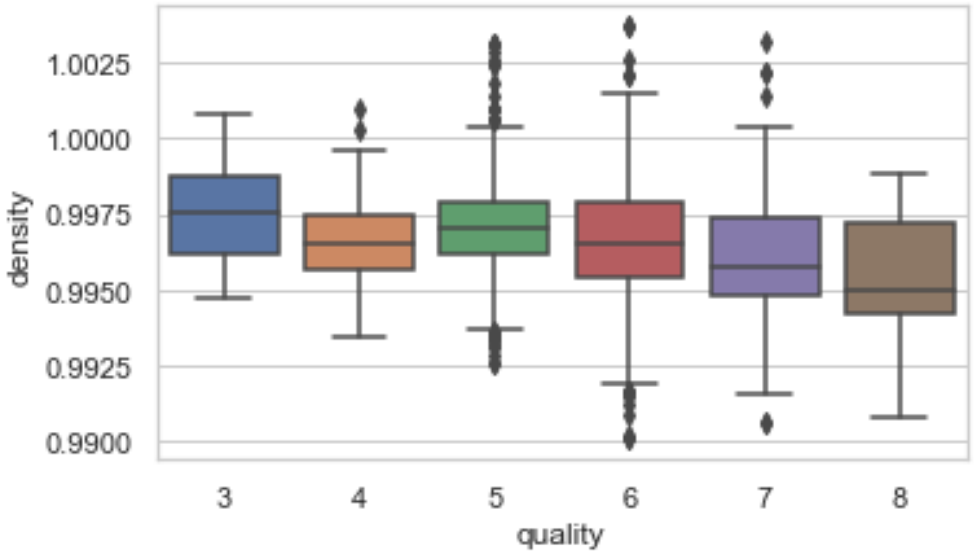

Figure 1.8. boxplot: quality - density

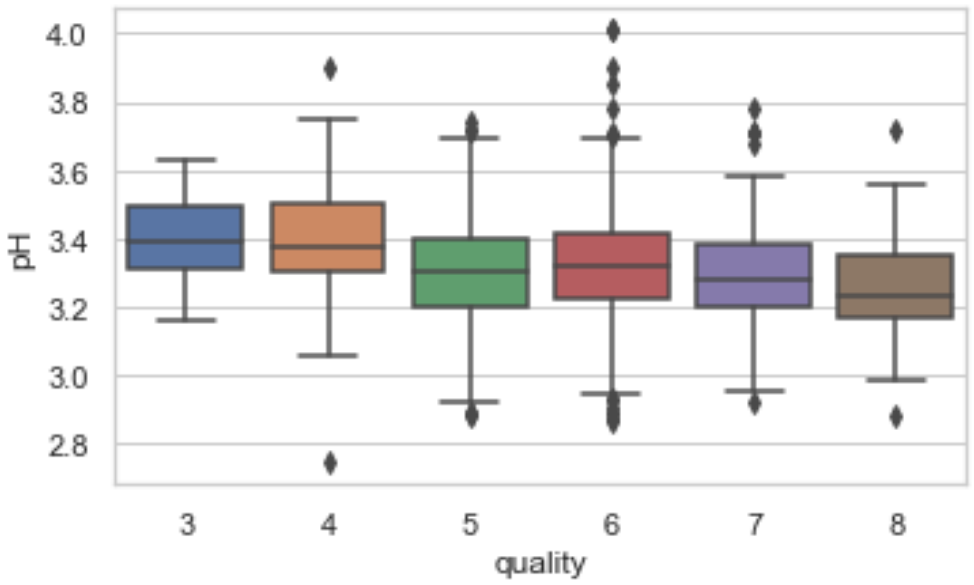

Figure 1.9. boxplot: quality - pH

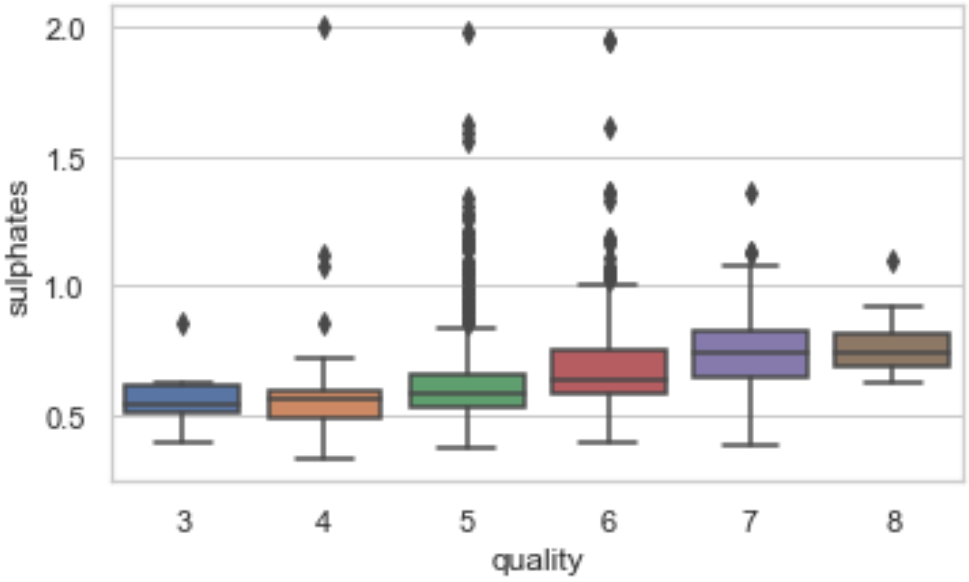

Figure 1.10. boxplot: quality - sulphates — This boxplot points out the complexity of getting an excellent result in terms of wine taste, without the use of sulphates, that for example are even avoided in the biodinamical cultures.

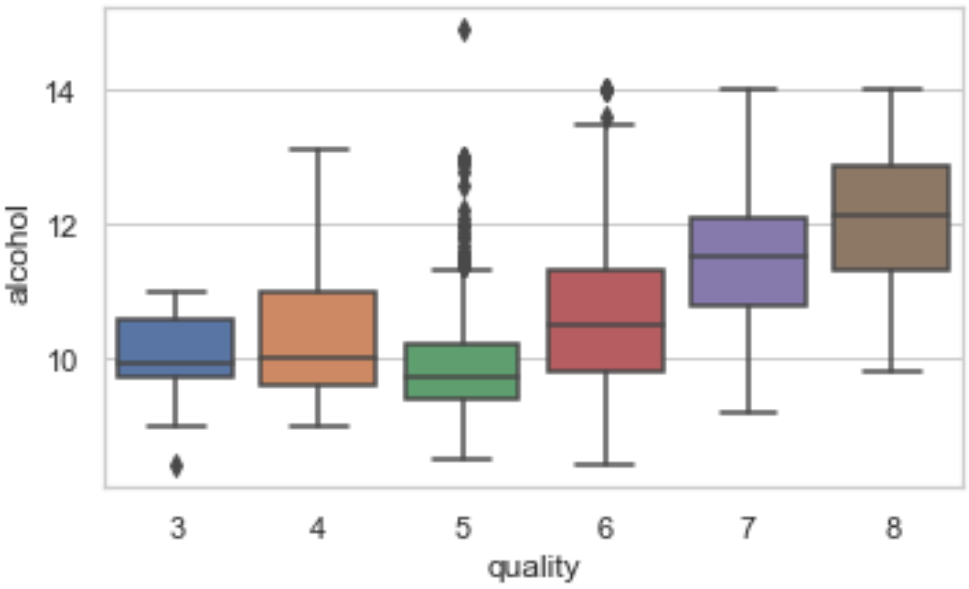

Figure 1.11. boxplot: quality - alcohol — Another evident peculiarity is that very well rated wines tend to have a higher alchohol content, or, more precisely, very 'light' wines are not good wines

Chapter 2

Unsupervised Learning

2.1 Unsupervised Learning

Before applying the supervised learning methods, where the goal is to predict Y (in our case the quality) using X1, X2...Xp (input features), we would like to apply PCA, hoping that it can be a useful preprocessor for supervised learning. In the unsupervised learning we're not interested in prediction, but we would like to discover interesting things about the measurements and visualize the data in order to find subgroups among the observations.

2.2 PCA

In our example, we drop out the qualitative feature 'quality' and we apply PCA and plot it: The

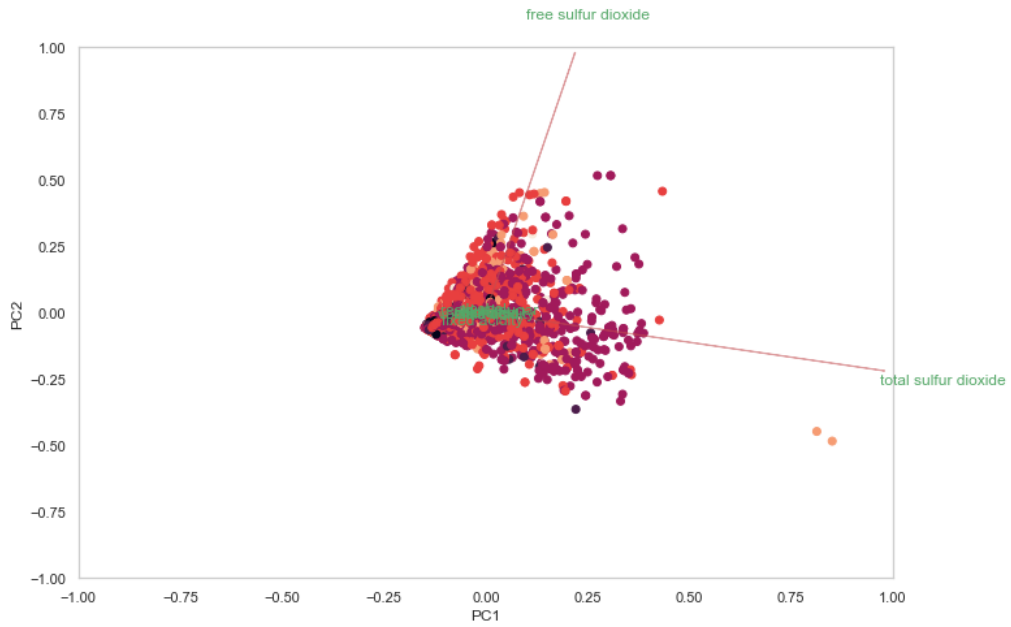

Figure 2.1. PCA

plot shows a dense concentration of most of the Loading Vectors in the same zone with similar values.

```
1st component, feature fixed acidity: -0.006132
2nd component, feature fixed acidity: -0.023899
1st component, feature volatile acidity: 0.000384
2nd component, feature volatile acidity: -0.002010
1st component, feature citric acid: 0.000171
2nd component, feature citric acid: -0.003035
1st component, feature residual sugar: 0.008649
2nd component, feature residual sugar: 0.011135
1st component, feature chlorides: 0.000064
2nd component, feature chlorides: -0.000237
1st component, feature free sulfur dioxide: 0.218857
2nd component, feature free sulfur dioxide: 0.975266
1st component, feature total sulfur dioxide: 0.975678
2nd component, feature total sulfur dioxide: -0.218917
1st component, feature density: 0.000004
2nd component, feature density: -0.000025
1st component, feature pH: -0.000268
2nd component, feature pH: 0.003272
1st component, feature sulphates: 0.000223
2nd component, feature sulphates: 0.000619
1st component, feature alcohol: -0.006358
2nd component, feature alcohol: 0.014564
```

Figure 2.2. PCA loading vectors - Coefficients

The exceptions, which are the most relevant coefficients, are:

- 1st component, feature ‘free sulfur dioxide’: 0.218857
- 2nd component, feature ‘free sulfur dioxide’: 0.975266
- 1st component, feature ‘total sulfur dioxide’: 0.975678
- 2nd component, feature ‘total sulfur dioxide’: -0.218917

We can assert that the 1st component roughly corresponds to a measure of the Total Sulfure Dioxide level, while the second component is more related to the amount of Free Sulfure Dioxide. In the first 2 PCs these two features are the only ones that show a big difference from the others in terms of Loading Vector amounts, so we can suppose that they are less correlated with the others. Moreover, we can make a hypotesis about a higher correlation among all the other variables. For this reason in the next analysis we will try to drop these two features.

Chapter 3

Supervised Learning

3.1 Supervised Learning

Supervised Learning approach is used to predict the response for future observations (prediction) or better understand the relationship between the response and the predictors (inference). In our case, we will compare some Supervised Learning algorithms that we apply with the aim to predict the quality of the wines, on the basis of the values of the quantitative variables. As a preliminar operation, we split our dataset in two parts: training and test data. The training data (usually most of the dataset) are used to train the algorithm, that will be tested with the remaining test data. The choice is 80 per cent of the dataset used for training data, 20 per cent for test data.

3.2 Resampling Methods

Resampling methods are computationally expensive methods that involve fitting the same statistical methods multiple times, using different subsets of the training data. Cross Validation can be applied to estimate the test error associated with a given statistical learning method in order to evaluate its performance, or to select the appropriate level of flexibility. The test error is the average error that results from using a statistical learning method to predict the response on a new observation.

3.2.1 K-fold Cross Validation

This approach involves randomly dividing the set of observations into k groups, or folds, of approximately equal size. The first fold is treated as validation set, and the method is fit on the remaining k-1 folds. The MSE is computed on the observations in the held-out fold. Repeating this computation k-times, holding out at each step a different fold, and averaging the MSE, we obtain the CV estimate. In our experiment, we will perform the 10-fold CV.

3.3 Linear Regression

The linear regression is one of the simplest approaches to supervised learning. Simple linear regression is a useful method for predicting a response on the basis of a single predictor variable. In our dataset, we have 10 quantitative variables, and we still have very few information about how they are correlated with one another. For this reason, it's more convenient to start from a Multiple Linear Regression analysis. Moreover, Multiple Linear Regression can be a useful support to interpret some misleading results that come from a Simple Linear Regression, that can suggest a direct causality between predictor and response, which is not real. In order to avoid the claim of causality, correlations among predictors cause problems, because the variance

of all coefficients tends to increase, sometimes dramatically. Moreover, interpretations become hazardous: when X_j changes, everything else changes.

The ideal scenario is when the predictors are uncorrelated, that is a balanced design:

- each coefficient can be estimated and tested separately
- interpretations such as "a unit change in X is associated with a change in Y, while all other variables remains fixed" are possible.

3.3.1 Multiple Linear Regression

The β coefficient of each variable quantifies the association between that variable and the response. We can interpret it as the average effect on Y of a unit increase in X_j, holding all other predictors fixed. The approach for estimating the coefficients is, as in linear regression, a least square computation. A relevant statistic in Multiple Linear Regression is the F-statistic. If our F-statistic is high, it suggests that at least one of the features is related to 'quality'. The R^2 tends to increase when more variables are added to the linear model, especially when there's correlation between the variables. In the last chapter, after PCA we made the hypothesis that 'Free Sulfure Dioxide' and 'Total Sulfure Dioxide' are the least correlated with the others. So, if we drop them, we'd expect a small decrease of R^2 value. RSE tends to increase if the decrease in RSS is small relative to the increase in p. The next figures show the computed coefficients, a sample of 25 observation compared with its predictions, and the correlation matrix, that compute all the correlations between each couple of variables.

	Coefficients	Standard Errors	t values	Probabilites
0	34.9987	11.327	3.090	0.002
1	0.0413	0.014	2.979	0.003
2	-1.1495	0.063	-18.160	0.000
3	-0.1779	0.079	-2.265	0.024
4	0.0279	0.008	3.514	0.000
5	-1.8734	0.221	-8.467	0.000
6	0.0027	0.001	2.308	0.021
7	-0.0028	0.000	-7.255	0.000
8	-31.5167	11.562	-2.726	0.006
9	-0.2545	0.103	-2.480	0.013
10	0.9240	0.060	15.490	0.000
11	0.2678	0.014	19.002	0.000

Figure 3.1. Multiple Linear Regression - Coefficients

R-squared:	0.365
Adj. R-squared:	0.360
F-statistic:	66.34
Prob (F-statistic):	6.26e-117

Figure 3.2. Multiple Linear Regression - F statistic

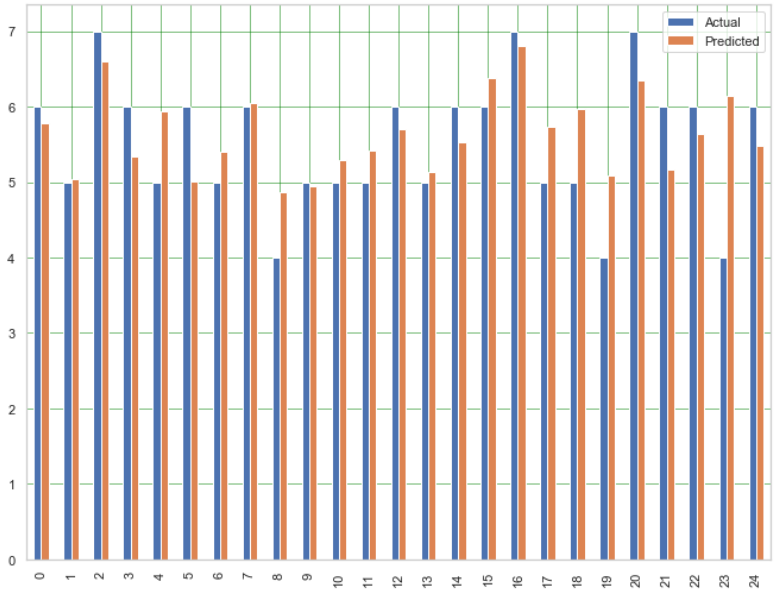

Figure 3.3. Multiple Linear Regression - Actual vs Predicted

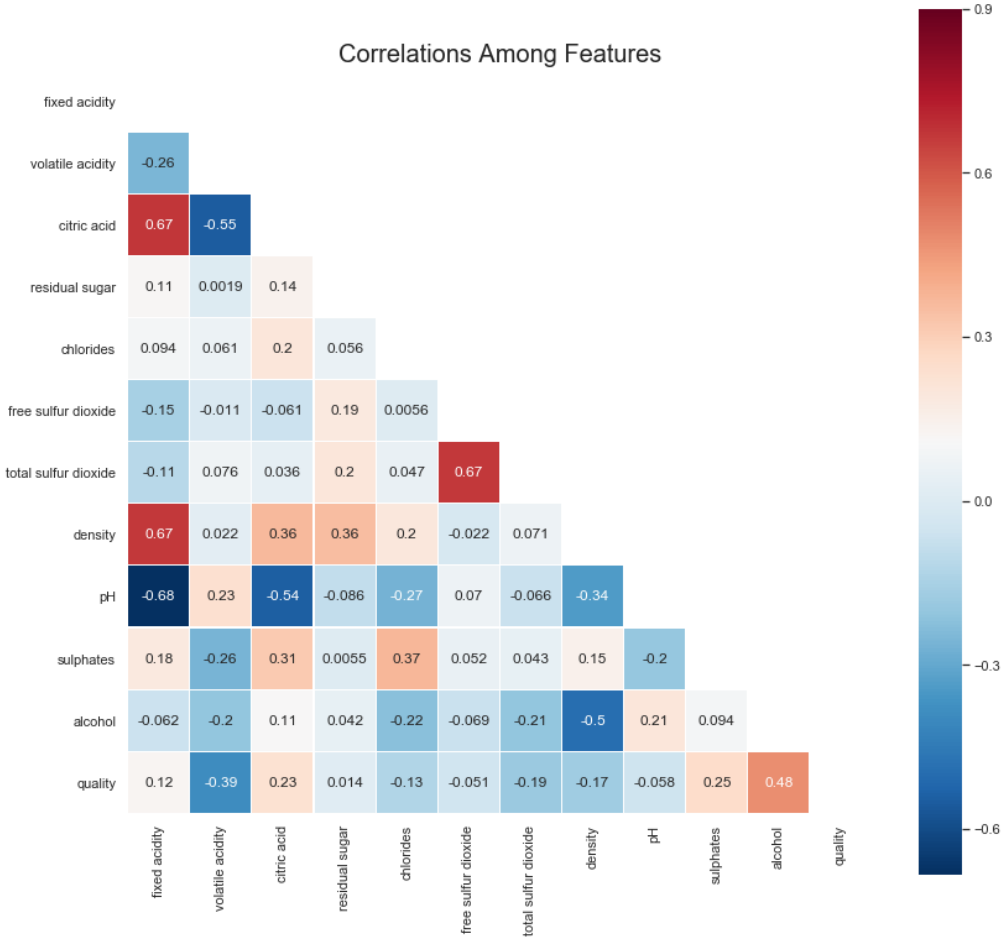

Figure 3.4. Correlation Matrix

From the correlation matrix (Figure 3.4) we can see that the highest values are related with 'fixed acidity'. In particular, 'fixed acidity' seems to be highly correlated both with 'citric acid' and 'density'. But if we see the correlation between those last two features ('citric acid' and 'density'), we can assume that they doesn't depend so much with one another, so we can consider it as a 'direct' relationship. The fact that 'fixed acidity' is correlated with the quantity of citric acid is not so interesting as the relationship between 'density' and 'fixed acidity'. Free and total sulfur dioxide are correlated with each other, but, as expected from the PCA, not with other feature (we will try later not to consider them). About 'quality', that will be the target in the classification analysis, we can see that the most influent features in linear terms are alcohol (best wines are generally more alcoholic), and volatile acidity that tends to decrease. Those are expected results. We can plot the simple linear regression with the most 'unexpected' relationship: the one between 'fixed acidity' and 'density', with its statistics.

3.3.2 Simple Linear Regression

The main statistics of the linear regression applied on 2 variables, are:

- RSS - residual standard error
- R^2 - correlation (linear relationship) between X and Y
- The t-distribution, that is the number of standard deviations that β_1 is away from 0. Similar to normal distribution for n $>$ 30.
- The p-value, the probability of observing any value equal to —t— or larger assuming β_1=0. A small p-value there is an association between the predictor and the response.

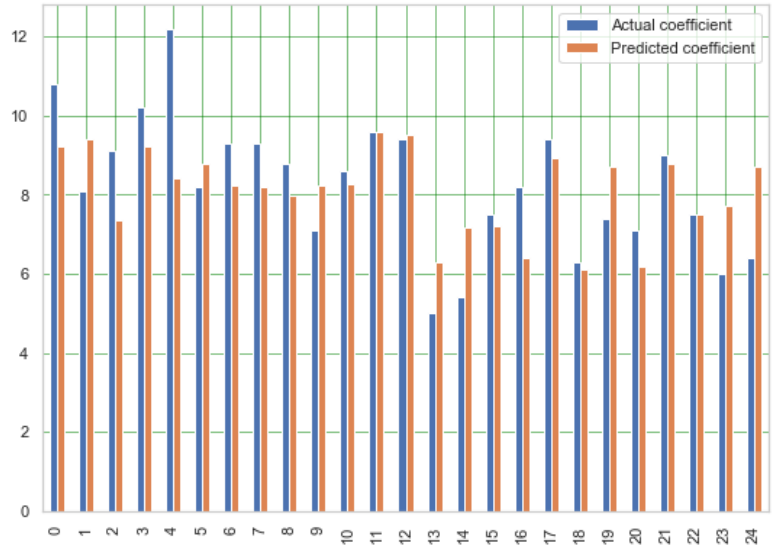

Figure 3.5. Simple Linear Regression - density-fixed acidity - Actual vs Predicted

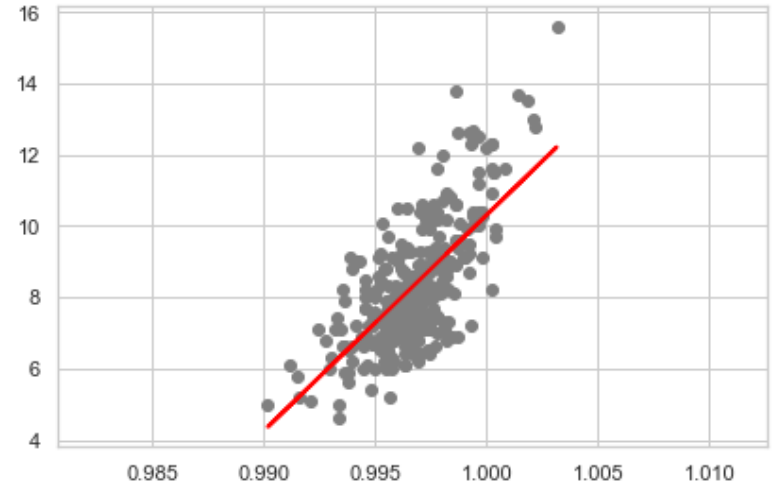

Figure 3.6. Simple Linear Regression- density-fixed acidity

	coef	std err	t	P>|t|	[0.025	0.975]
const	-596.0537	19.360	-30.788	0.000	-634.034	-558.073
x1	606.3572	19.423	31.219	0.000	568.253	644.461

Figure 3.7. Simple Linear Regression- density-fixed acidity

3.4 Classification

Our database is very suitable for classification problems, because we have just one qualitative variable, 'quality' which is composed of 9 discrete values (but in our dataset there are no wines with quality values of 1, 2, 9). Predicting qualitative response is a typical approach of classification, and it consists of assigning an observation into a category. We can simplify the categorical distintion splitting the quality in 'good' and 'bad': good wines if quality >6, bad wines if <=6. The amount of good wines is lower than bad wines.

The classification report function that we will use, provides scores about:

- Precision - the ability of a classifier not to label an instance positive that is actually negative. For each class it is defined as the ratio of true positives to the sum of true and false positives. Said another way, "for all instances classified positive, what percent was correct?"
- Recall - the ability of a classifier to find positive instances. For each class it is defined as the ratio of true positives to the sum of true positives and false negatives. Said another way, "for all instances that were actually positive, what percent was classified correctly?"
- F1 score - a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0. Generally speaking, F2 scores are lower than accuracy measures as they embed precision and recall into their computation. As a rule of thumb, the weighted average of F1 should be used to compare classifier models, not global accuracy. We'll show it in the beginning of our classification reports as "Score: ".
- Support - the number of actual occurrences of the class in the specified dataset. Imbalanced support in the training data may indicate structural weaknesses in the reposted scores of the classifier and could indicate the need for stratified sampling or rebalancing. Support doesn't change between models but instead diagnoses the evaluation process.

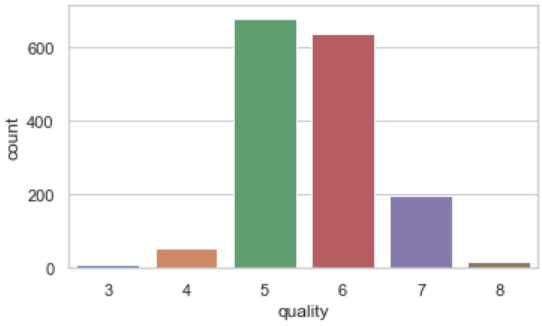

Figure 3.8. Quality - count

The reported averages include macro average (averaging the unweighted mean per label), weighted average (averaging the support-weighted mean per label), sample average (only for multilabel classification) and micro average (averaging the total true positives, false negatives and false positives) it is only shown for multi-label or multi-class with a subset of classes because it is accuracy otherwise.

3.4.1 KNN

Given a positive integer K, and a test observation x, the KNN classifier first identifies the K points in the training data that are closest to x. Then it estimates the conditional probability for a certain class, as the fraction of points that belongs to that class. It applies the Bayes rule and classifies the test observation to the class with the largest probability. The choice of K has a drastic effect on the classifier. When K is 1, the decision boundary is overly flexible and finds patterns in the data that don't correspond to the Bayes decision boundary. This corresponds to a classifier that has low bias but very high variance. As K grows, the method becomes less flexible and produces a decision boundary that is close to linear. This corresponds to a low-variance but high-bias classifier.

```
KNEIGHBORS CLASSIFIER 1.............................
Score:  0.853125
              precision    recall  f1-score   support

           0       0.91      0.92      0.91       273
           1       0.50      0.47      0.48        47

    accuracy                           0.85       320
   macro avg       0.70      0.69      0.70       320
weighted avg       0.85      0.85      0.85       320

KNEIGHBORS CLASSIFIER 5.............................
Score:  0.85625
              precision    recall  f1-score   support

           0       0.88      0.97      0.92       273
           1       0.53      0.21      0.30        47

    accuracy                           0.86       320
   macro avg       0.70      0.59      0.61       320
weighted avg       0.83      0.86      0.83       320

KNEIGHBORS CLASSIFIER 10............................
Score:  0.840625
              precision    recall  f1-score   support

           0       0.85      0.98      0.91       273
           1       0.17      0.02      0.04        47

    accuracy                           0.84       320
   macro avg       0.51      0.50      0.48       320
weighted avg       0.75      0.84      0.78       320
```

Figure 3.9. [1, 5, 10]-Nearest Neighbors - classification reports

```
KNEIGHBORS CLASSIFIER 50............................
Score:  0.853125
              precision    recall  f1-score   support

           0       0.85      1.00      0.92       273
           1       0.00      0.00      0.00        47

    accuracy                           0.85       320
   macro avg       0.43      0.50      0.46       320
weighted avg       0.73      0.85      0.79       320

KNEIGHBORS CLASSIFIER 100...........................
Score:  0.853125
              precision    recall  f1-score   support

           0       0.85      1.00      0.92       273
           1       0.00      0.00      0.00        47

    accuracy                           0.85       320
   macro avg       0.43      0.50      0.46       320
weighted avg       0.73      0.85      0.79       320
```

Figure 3.10. [50, 100]-Nearest Neighbors classification reports

3.4.2 Logistic Regression

Many aspects of logistic regression are similar to the linear regression. Rather than modeling a response directly, logistic regression models the probability that Y belongs to a particular category: p(X) = Pr(Y=1|X). In order to model that probability with a function that gives outputs between 0 and 1, the linear regression is no more adapt. Logistic regression uses instead the 'logistic function', and to fit the model (to estimate the coefficients) the 'maximum likelihood'. The maximum likelihood seeks estimates for the coefficients such that the predicted probability of a category value, using the logistic function, corresponds as closely as possible to the individual's observed category value. In mathematical terms, it means finding the coefficients that maximize the likelihood function, and plugging them into the logistic function.

3.4.3 LDA - Bayes Theorem For Classification

With this approach, we model the distribution of the predictors X separately in each of the response classes (i.e. given Y), and then use Bayes theorem to flip these around into estimates Pr(Y=1|X=x). LDA has the same form as logistic regression, the difference is in how the parameters are estimated. In fact logistic regression uses the conditional likelihood (discriminative learning), while LDA uses the full likelihood based on Pr(X,Y) (generative learning). Despite these differences, in practice the results are often very similar. When these distributions are normal the model is very similar to logistic regression. Instead of LR, LDA works well where the classes are well separated, when n is small and the distribution of the predictors X is approximately normal in each of the classes, and when we have more than 2 response classes.

```
LOGISTIC REGRESSION CLASSIFIER.............................
Score:  0.865625
              precision    recall  f1-score   support

           0       0.88      0.98      0.93       273
           1       0.62      0.21      0.32        47

   micro avg       0.87      0.87      0.87       320
   macro avg       0.75      0.60      0.62       320
weighted avg       0.84      0.87      0.84       320

LOGISTIC REGRESSION WITH CV
Score:  0.8803949311023622
```

Figure 3.11. Logistic Regression - classification report

```
LDA............................
Score:  0.85
              precision    recall  f1-score   support

           0       0.89      0.93      0.91       273
           1       0.49      0.36      0.41        47

   micro avg       0.85      0.85      0.85       320
   macro avg       0.69      0.65      0.66       320
weighted avg       0.83      0.85      0.84       320

LDA WITH CV
Score:  0.8850578248031497
```

Figure 3.12. LDA - classification report

3.4.4 Tree Based Methods

The tree based methods are used for regression and classification. They involve stratifying or segmenting the predictor space into a number of simple regions. In order to make a prediction for a given observation, they typically use the mean or the mode of the training observations in the region to which it belongs.

Decision Trees

The decision tree is simple and it's useful for interpretation, as a matter of fact it mirrors more closely the human decision-making than, for example, linear regression. Unfortunately it typically is not competitive with the the best supervised learning approaches in terms of prediction accuracy. The tree building process take a top-down greedy approach with a recursive binary splitting of the data, since it is not computationally feasible to consider every possible partition of the feature space in order to find the optimal solution.

Random Forest

Bagging is a procedure for reducing the variance of a statistical learning method. Averaging a set of observations, instead of a single one, reduces the variance. That's not a pratical approach since we don't have multiple training sets, but we can take repeated samples from the single training dataset. We average all the predictions obtained by different bootstrapped datasets. Random forests is an improvement over bagged trees, with a small tweak that decorrelates the trees. For this reason we're expecting a more accurate classification.

```
DECISION TREE.........................
Score:  0.871875
              precision    recall  f1-score   support

           0       0.92      0.93      0.93       273
           1       0.57      0.53      0.55        47

    accuracy                           0.87       320
   macro avg       0.74      0.73      0.74       320
weighted avg       0.87      0.87      0.87       320

DECISION TREE WITH CV...................
Score:  0.8741449311023622
```

Figure 3.13. Decision Tree - classification report

```
RANDOM FOREST
Score:  0.896875
              precision    recall  f1-score   support

           0       0.92      0.96      0.94       273
           1       0.69      0.53      0.60        47

   micro avg       0.90      0.90      0.90       320
   macro avg       0.81      0.75      0.77       320
weighted avg       0.89      0.90      0.89       320

RANDOM FOREST WITH CV
Score:  0.9093134842519686
```

Figure 3.14. Random Forest - classification report

3.4.5 Support Vector Machine

The SVM is an approach for classification based on the concept of hyperplane: in a p-dimensional space, a hyperplane is a flat affine subspace of dimension p-1. In order to separate data in the best way, among the possible separating hyperplanes, the support vector classifier chooses the

one that is the farthest from the training observations. Then we can classify a test observation based on which side of the maximal margin hyperplane it lies. If a separating hyperplane doesn't exist, we can add a tolerance rate for mislabeled observations. The C parameter is the tuning parameter, and it determines the number and the severity of the violations to the margin that we tolerate, and it can be chosen via Cross Validation. The SVM converts this approach for building non linear decision boundaries, using kernels. The kernel is a function that quantifies the similarity of two observation, and it actually corresponds to the inner product, that in the linear case, provided a measuerment of distance. In our expreriment we try to find the best C value, in order to find the best classification. The default C value in "sklearn.svm.SVC()" function is 1. Trying with: 0.001, 0.01, 0.1, 1, 10, 100, 1000, we obtain that the default value (C=1) provides the best accuracy score, with and without CV.

```
SVM WITH CV C=0.001000
Score:  0.8670829232283465
SVM NO CV C=0.010000
Score:  0.853125
SVM WITH CV C=0.010000
Score:  0.8670829232283465
SVM NO CV C=0.100000
Score:  0.853125
SVM WITH CV C=0.100000
Score:  0.8670829232283465
SVM NO CV C=1.000000
Score:  0.85
SVM WITH CV C=1.000000
Score:  0.8889886811023622
SVM NO CV C=10.000000
Score:  0.84375
SVM WITH CV C=10.000000
Score:  0.8710199311023622
SVM NO CV C=100.000000
Score:  0.84375
SVM WITH CV C=100.000000
Score:  0.8702448326771653
SVM NO CV C=1000.000000
Score:  0.853125
SVM WITH CV C=1000.000000
Score:  0.8632012795275591
```

Figure 3.15. SVM - different values of C

```
SVM.................
Score:  0.85
              precision    recall  f1-score   support

           0       0.87      0.97      0.92       273
           1       0.46      0.13      0.20        47

    accuracy                           0.85       320
   macro avg       0.66      0.55      0.56       320
weighted avg       0.81      0.85      0.81       320
```

Figure 3.16. SVM - classification report

Chapter 4

Conclusions

4.0.1 Classification methods' comparison

The first consideration is about the classification split that we chose. Even if the accuracy scores look very high, as the distribution of the 2 output is strongly imbalanced, with a majority of 'bad wines', if we consider the whole classification report we can see that the predictions for the value 1 (good) are not very precise. Anyway, we can have an idea about which algorithm works better and why. Generally, when the classes are well separated, SVM and LDA tend to behave better than logistic regression. In our case, logistic regression provides a better result than LDA and SVM, so we can assert that the data are quite overlapped. Random Forest classifier works better than a simple Decision Tree, and actually it's the only one that gives satisfactory results even for the prediction of good wines. KNN is working better when the K parameter is low, and that's another evidence about the non-linear separation of the two classification. Moreover, as expected, Cross Validation had a positive effect on accuracy with all the used algorithms.

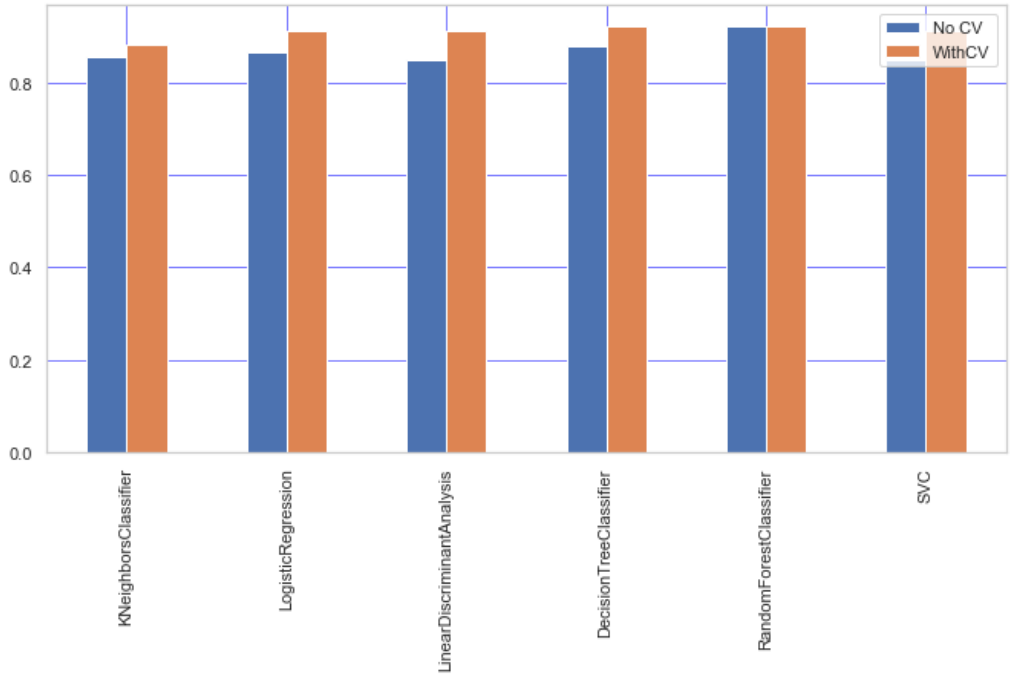

Figure 4.1. Accuracy score summary

4.0.2 Removing two variables

As we said, we tried to apply the same supervised learning algorithms, with a 'cutted' dataset. We removed the 2 quantitative variables that look less correlated with the PCA. These are the results compared with the previous ones, with and without Cross Validation. The R squared statistic computed with the Multiple Linear Regression has strongly decreased, as expected. About the other algorithms, we found that accuracy scores are very similar, with some improvements and some worsenings. So, even if those 2 variables are not correlated with the others, they don't have a bad influence on predictions.

R-squared:	0.230
Adj. R-squared:	0.225
F-statistic:	42.22
Prob (F-statistic):	2.23e-66

Figure 4.2. Multiple Linear Regerssion with cutted db

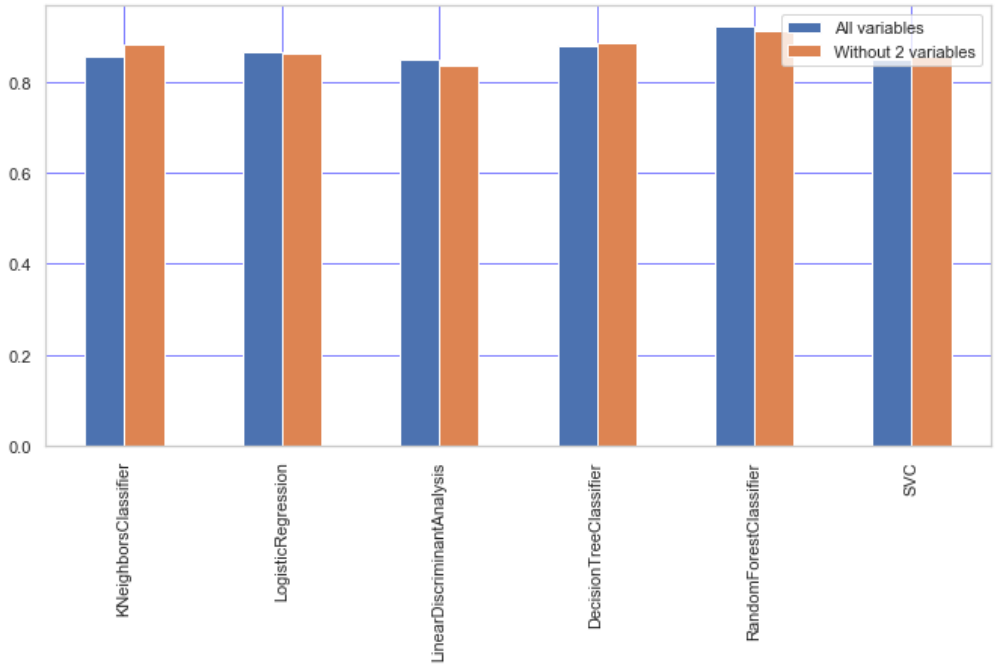

Figure 4.3. Accuracies comparison with cutted db - without CV

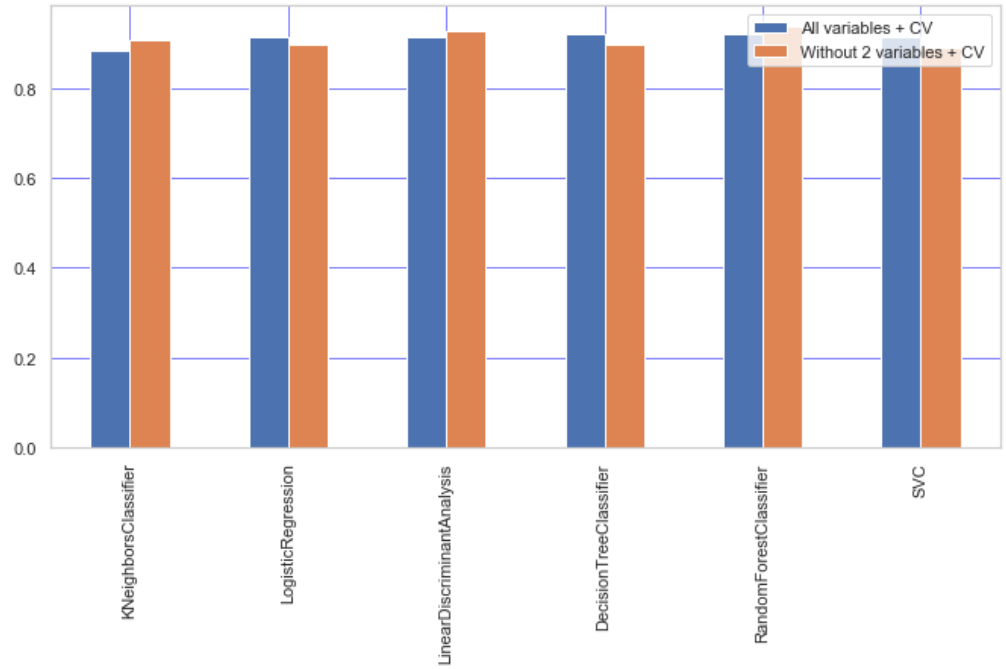

Figure 4.4. Accuracies comparison with cutted db - with CV

4.0.3 More balanced classification

If we split the dataset in 2 more balanced parts, with good wines when quality is at least 6 and bad when the rating is insufficient, then we can see that even if the overall accuracies are lower, the predictions for good (or not bad) wines are more precise. Random forest keeps on being the best algorithm. KNN is not working very well, and among the chosen value for K, 50 is the best, so with this classification choice, the decision boundary tends to be more linear. Even here, the cutted database doesn't provide improvements. We show the classification report for these two algorithms, and the histograms with the final comparisons.

```
KNEIGHBORS CLASSIFIER 50.................................
Score:  0.65625
              precision    recall  f1-score   support

           0       0.61      0.60      0.61       141
           1       0.69      0.70      0.69       179

    accuracy                           0.66       320
   macro avg       0.65      0.65      0.65       320
weighted avg       0.66      0.66      0.66       320
```

Figure 4.5. Balanced split - KNN

RANDOM FOREST
Score: 0.8

	precision	recall	f1-score	support
0	0.77	0.78	0.77	141
1	0.82	0.82	0.82	179
accuracy			0.80	320
macro avg	0.80	0.80	0.80	320
weighted avg	0.80	0.80	0.80	320

RANDOM FOREST WITH CV
Score: 0.8210498783037294

Figure 4.6. Balanced split - Random Forest

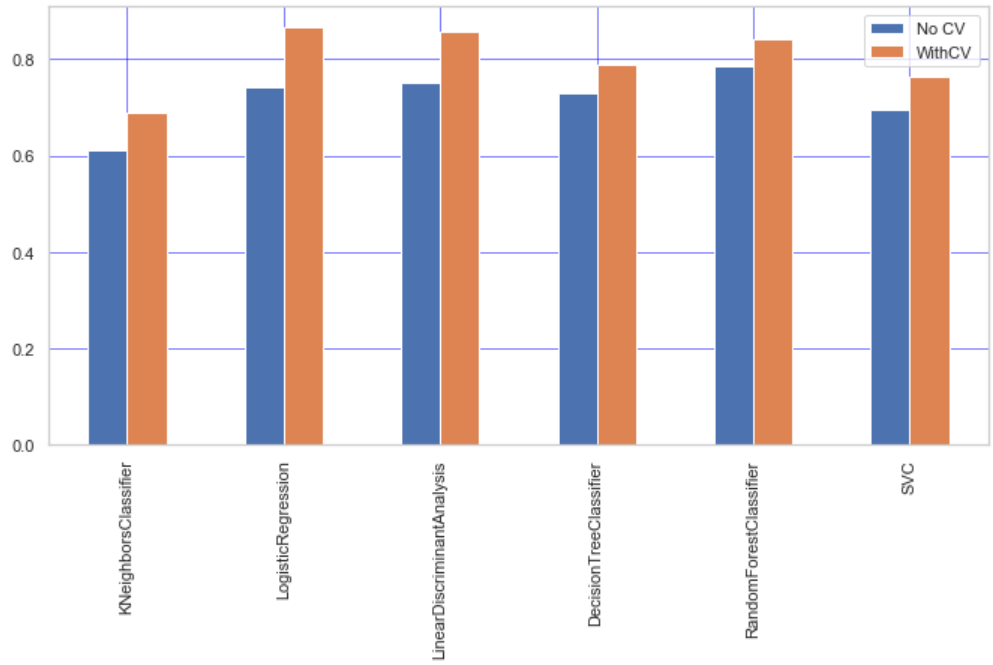

Figure 4.7. Balanced split - Accuracies comparisons

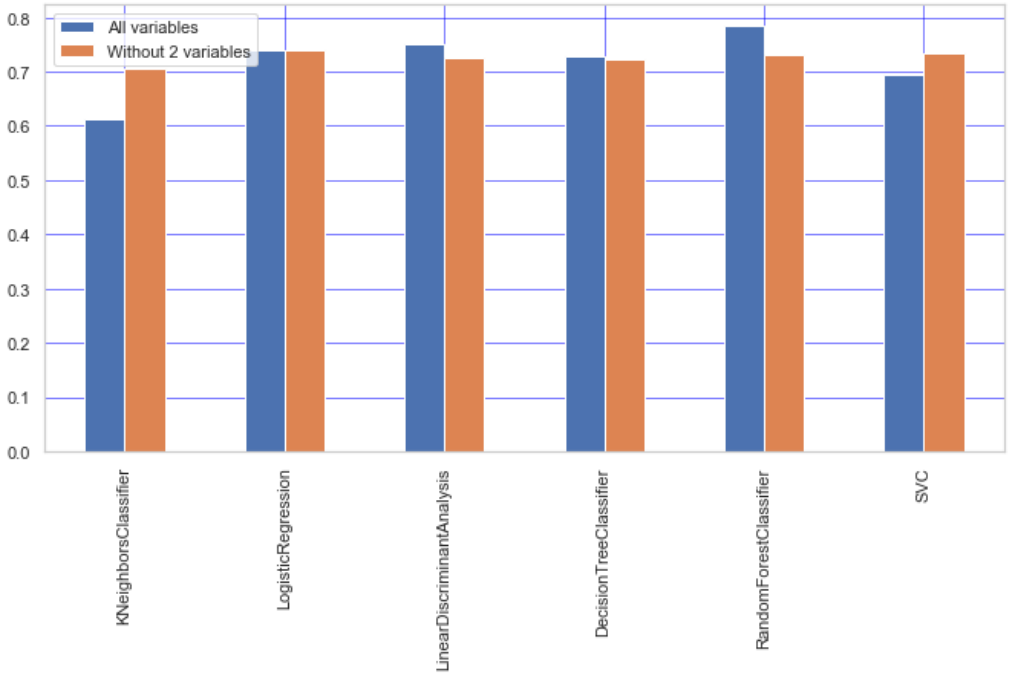

Figure 4.8. Balanced split - Accuracies comparison with cutted db

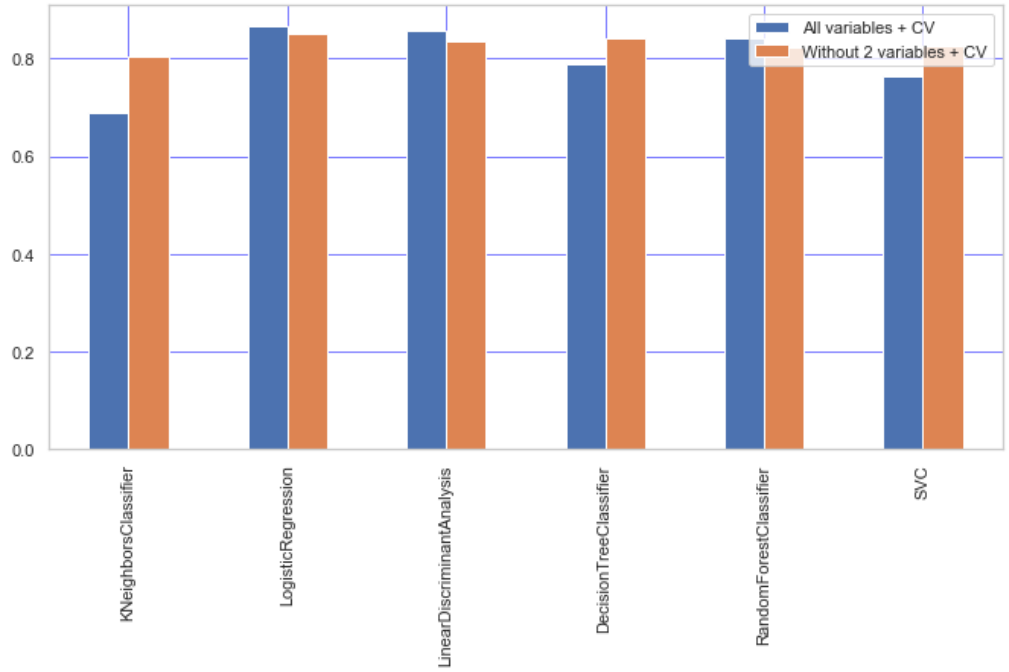

Figure 4.9. Balanced split - Accuracies comparison with cutted db - with CV

Bibliography

[1] G. James, D. Witten, T. Hastie, R.Tibshirani "An Introduction To Statistical Learning" edited by Springer Science+ Business Media New York 2013, 2013, DOI 10.1007/978-1-4614-7138-7