**Report of the project for the exam of the *Machine learning and pattern recognition* course.**

Candidate 1: Diego Gasco s296762

Candidate 2: Giovanni Genna s304684

**Abstract**

We have chosen to work on the *Wine quality detection* task. The request of the problem is to discriminate between good and bad quality wines. The first goal of our work is to study and analyse the provided problem, in particular the kind of features, their ranges and their distribution. The second part consists in developing the most appropriate classification algorithms and discarding models that are not proper for the considered task, by means of the training data. Finally, the different approaches chosen are evaluated on the test set.

**The dataset**

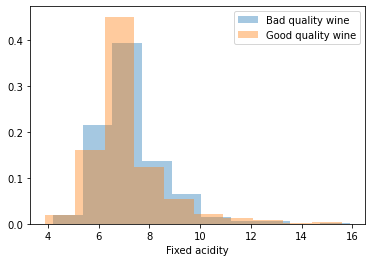
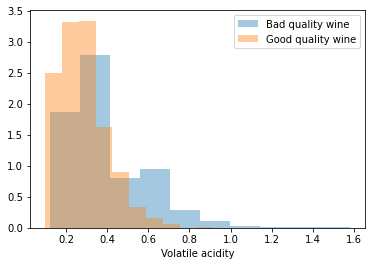
The dataset is taken from the UCI repository. The original dataset consists of 10 classes (quality 1 to 10). For this project, the dataset has been binarized, collecting all wines with low quality (score lower than 6) into class 0, and good quality (score greater than 6) into class 1. Wines with quality 6 have been discarded to simplify the task. The dataset contains both red and white wines (originally separated, they have been merged). There are 11 features, that represent physical properties of the wine:

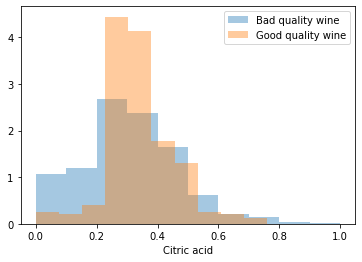
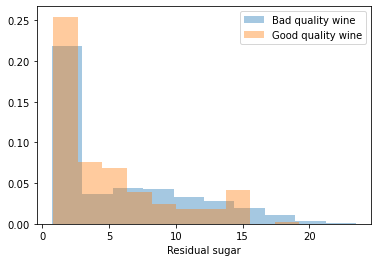
1. Fixed acidity
2. Volatile acidity
3. Citric acid
4. Residual sugar
5. Chlorides
6. Free sulfur dioxide
7. Total sulfur dioxide
8. Density
9. pH
10. Sulphates
11. Alcohol

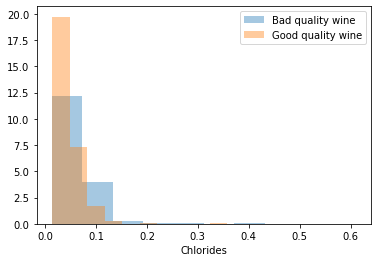
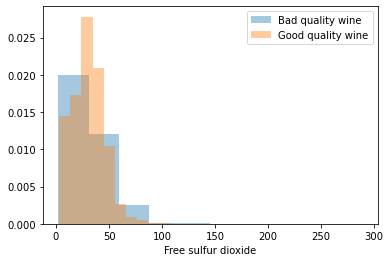
In the training set, there are 1839 samples, with their own features and another field with the class to which they belong (0 or 1). We modified the initial ‘Train.txt’ file to extract a matrix with on each column a sample (the numerical values of the eleven features: 11 rows, 1839 columns) and an array whose each element is the class label of the considered sample (we did these operations also for the ‘Text.txt’ file).

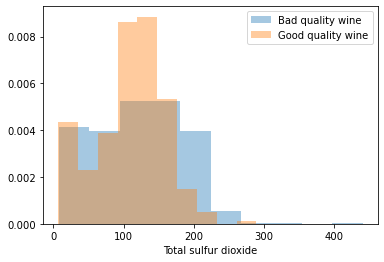
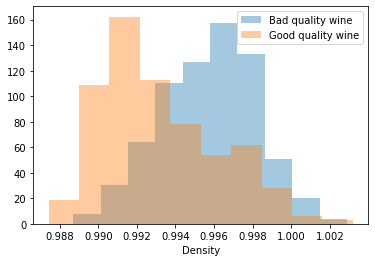
From this point, we will obviously consider only the data of the training set. Later, when we will start the evaluation of the models, the use of the test set will be specified.

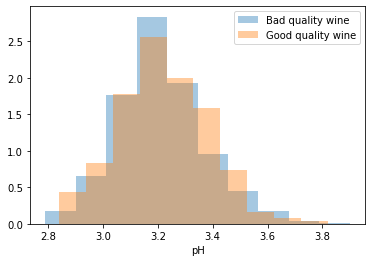
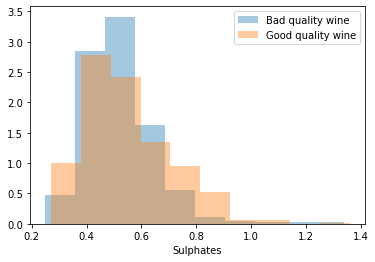
To understand how the several features of the two classes are distributed, for each of them, it is possible to plot the corresponding histogram:

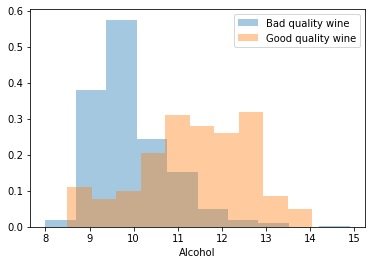
 

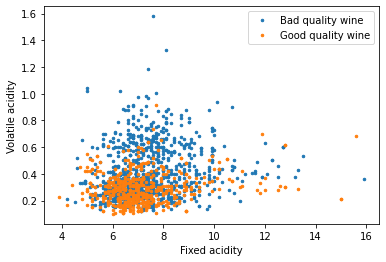
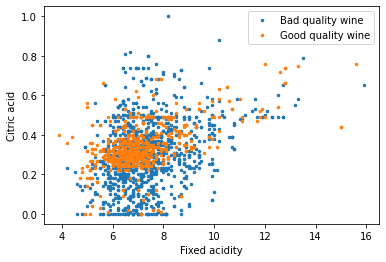
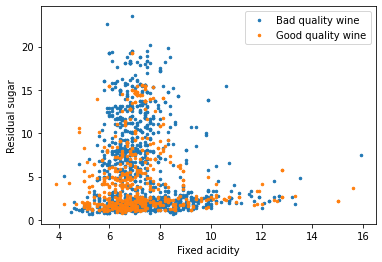
 

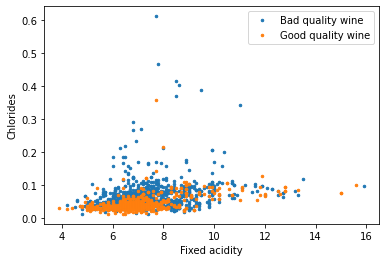
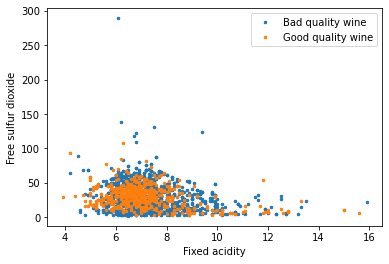
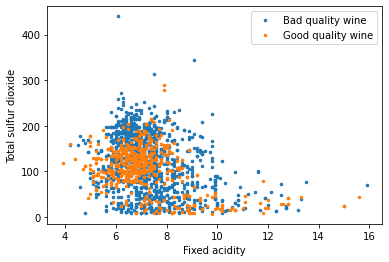
 

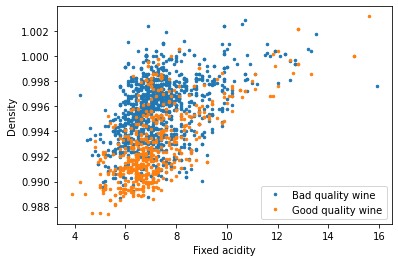
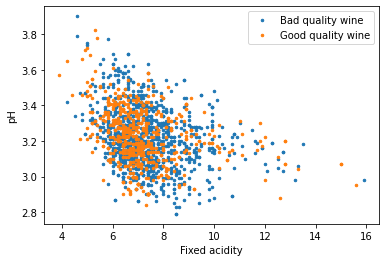
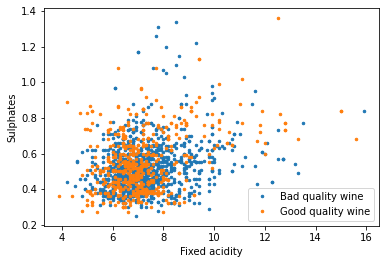


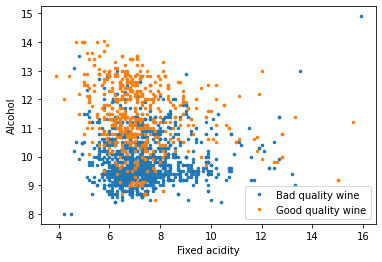
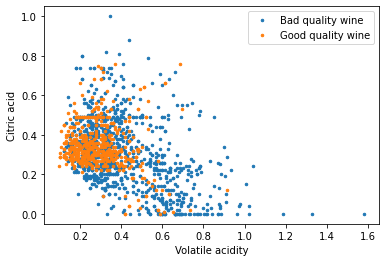
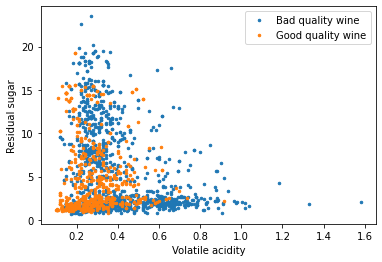
We can observe that, in general, for all features, there is no very well separation, vice versa, these is a large overlap and, sometimes, as in the case of *fixed acidity, residual sugar, pH* and *sulphates* the data is almost totally overlapping. For *density* and *alcohol*, it is possible to notice quite clearly that they are the features more separate. We can note that the data of the two classes for most of the features are distributed in a way similar to a Gaussian distribution, particularly for *fixed acidity, citric acid, density, pH, sulphates* and *alcohol*.

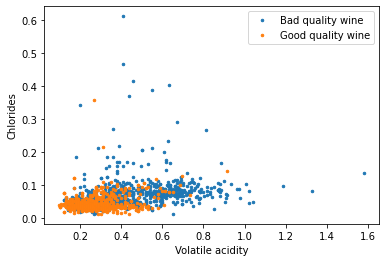
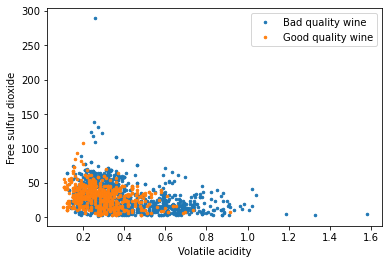
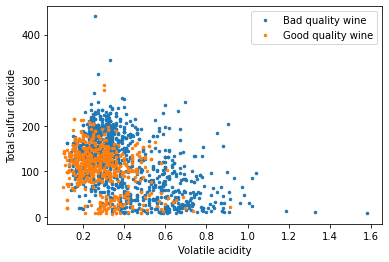
To have a view of the relationship between the different features for the different class, we can visualize the scatter plots of the different features pairs for each class:

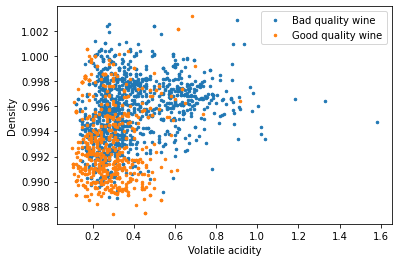
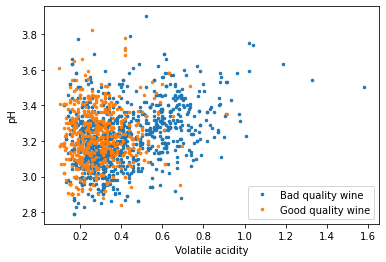
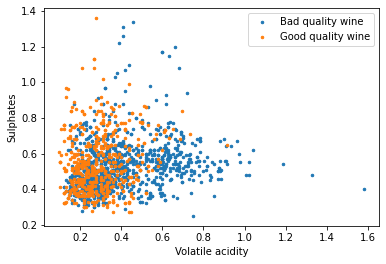
  

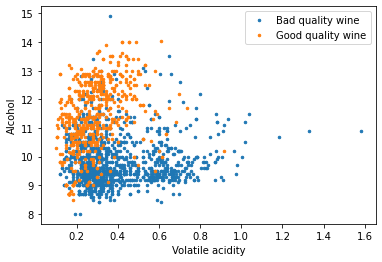
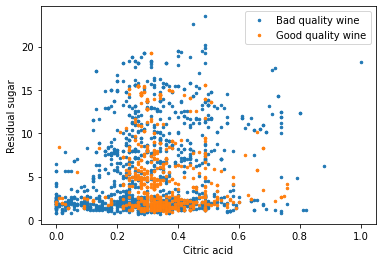
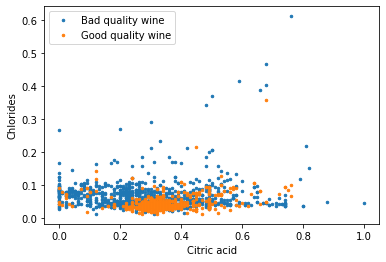
  

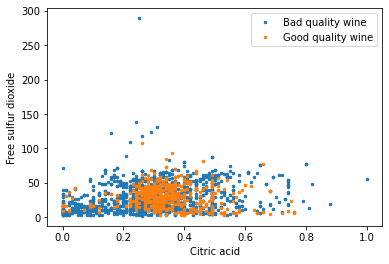
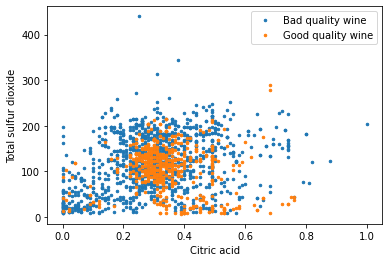
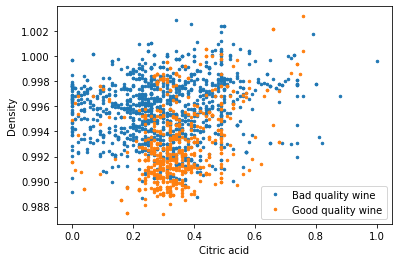
  

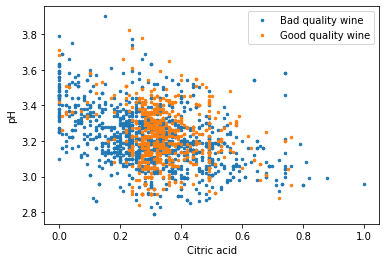
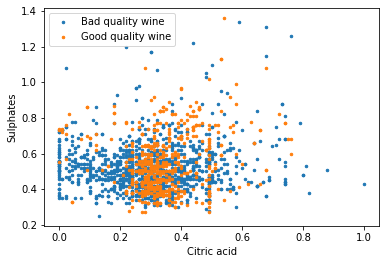
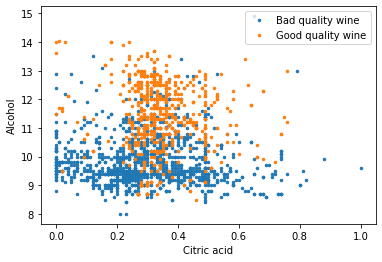
  

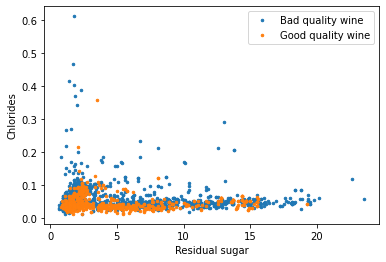
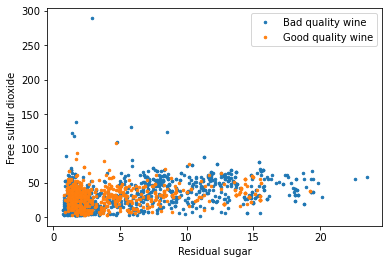
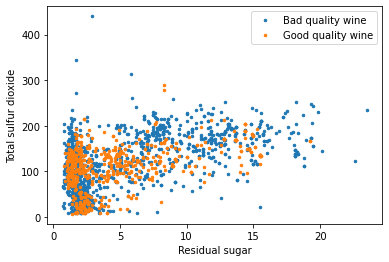
  

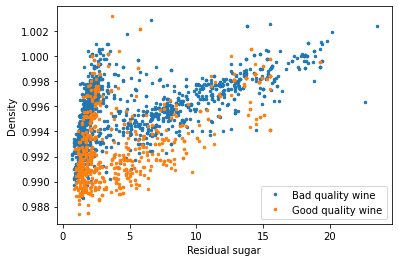
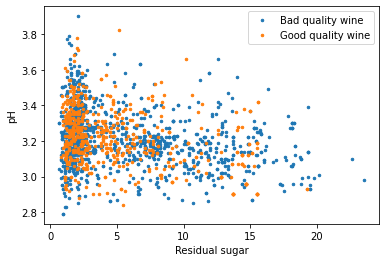
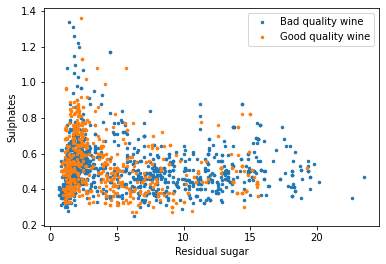
  

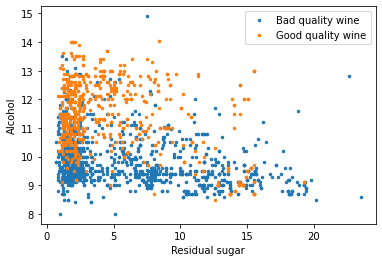
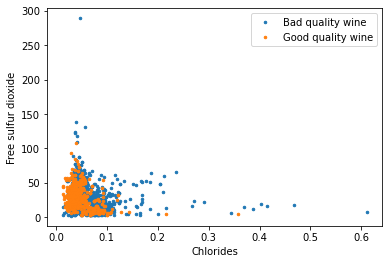
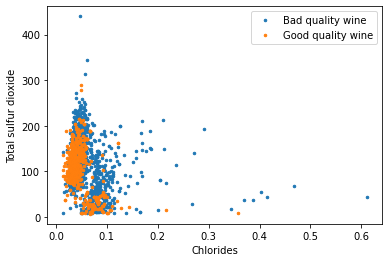
  

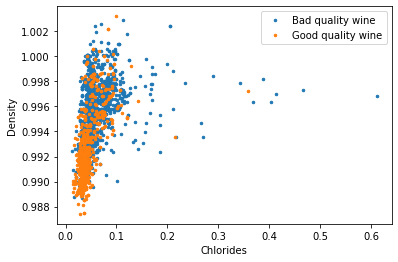
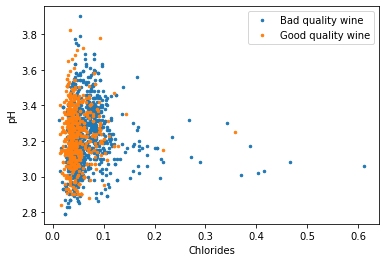
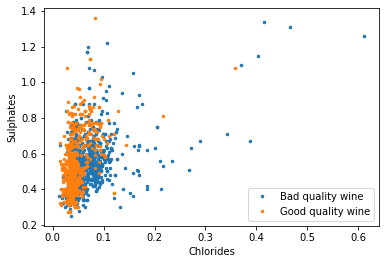
  

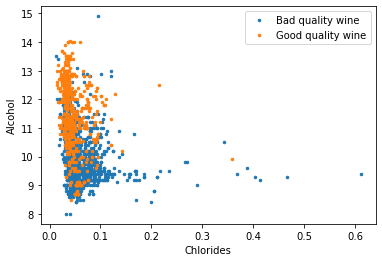
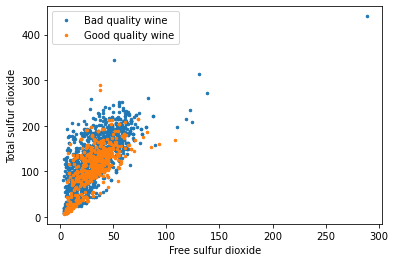
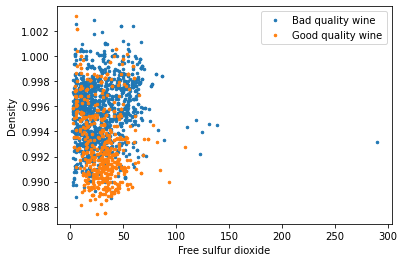
  

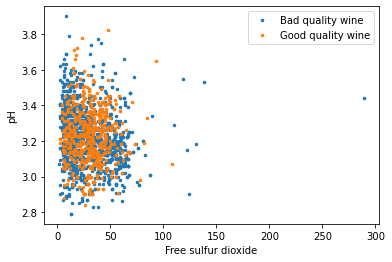
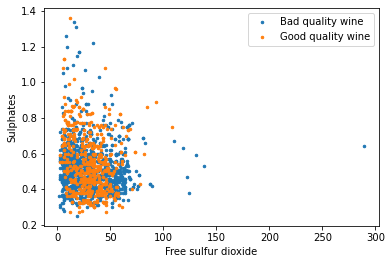
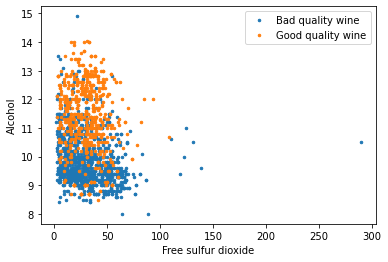
  

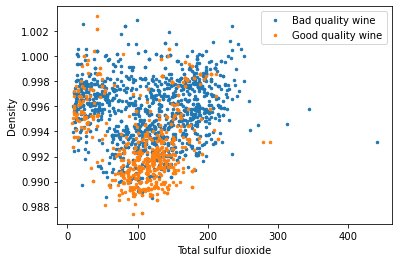
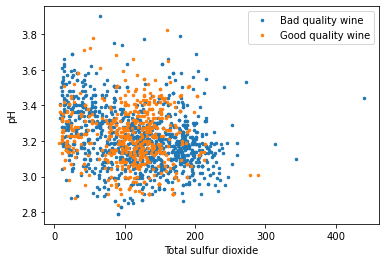
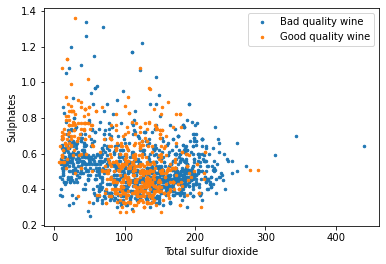
  

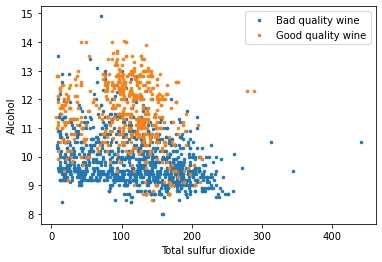
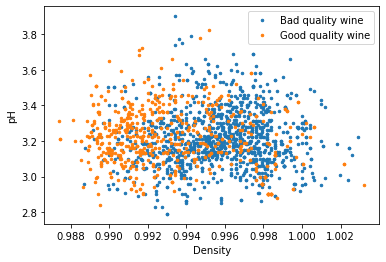
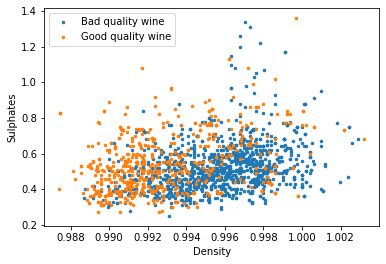
  

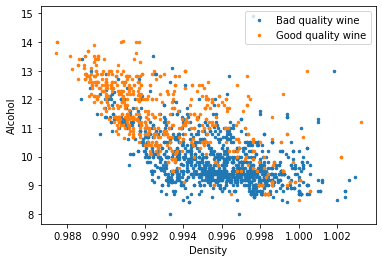
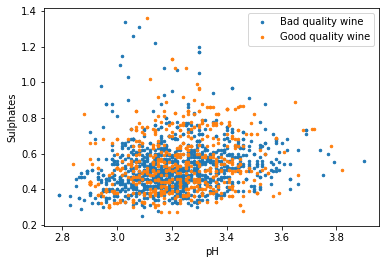
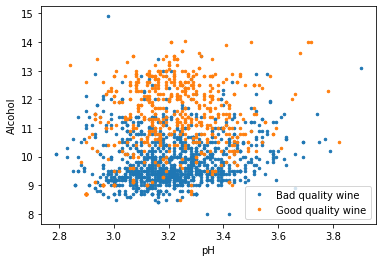
  

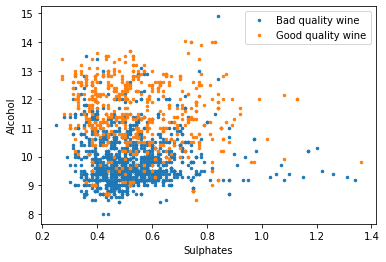
  



**Dimensionality Reduction**

Dimensionality reduction techniques compute a mapping from the n-dimensional original features space (in this case, n = 11), to a m-dimensional space, with m<n. The goal of these methods are several: compress information (also to reduce the computational time), remove unwanted variability, simplify classification, data visualization (in case of m = 2 and m =3).

11 features are not few, so it may be reasonable to apply a dimensionality reduction algorithm, to compress the information, preserving the most useful for the classification. The method we adopted for this problem is the *PCA (Principal Component Analysis)*, a linear unsupervised technique that finds a subspace of ℝn . *PCA* projects the data over the principal components. These can be computed from the eigenvectors of the data covariance matrix *C* corresponding to the largest eigenvalues.

*C =*

Where *N* is the number of samples, *xi*is the *i*-th sample and is the dataset mean:

*C* can be decomposed as:

*C= UΣUT*

Where *U* is a matrix, whose columns are the eigenvectors of *C* and *Σ* is a diagonal matrix, containing the eigenvalues in descending order.

*U = [u1 … um, um+1 … un]*

*P = [u1 … um]*

*P* corresponds to the first m columns of *U*.

Finally, we can apply the projection to the initial n-dimensional matrix of samples, to obtain the dimensionality reduction and have a m-dimensional matrix.

*Y = PTX*

Through experimental tests, we can affirm that, for this problem, the best results are obtained for m in the range [7, 9].

**Classifications**

It is not possible to use *LDA (Linear Discriminant Analysis)* as linear dimensionality reduction method, because it allows estimating at most C-1 directions, where C is the number of classes, but in this case, C is equal to 2. *LDA* can be used as a classification method for binary problems, so it could be applied to this problem.

…