# Machine Learning Project

Simone Grimaldi S260770
Alfredo Baldó Chamorro S288347

July 6, 2021

# Contents

## 1.1 Dataset

The dataset contains the information of red and white variants of the Portuguese "Vinho Verde" wine. It has been split into Test and Train sets containing 1822 and 1839 samples respectively. Moreover, there are two classes: good quality (value 1) and bad quality wine (value 0), where each has twelve attributes. Moreover, the table 1.1 resumes the information related to the attributes.

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

### 1.1.1 Correlations

The range of the each attribute is stated in the table 1.1. Moreover, the image 1.2 illustrates the correlation between the attributes in the dataset. It can be extracted from the figure that the most correlated attributes are (7,10), (6,5) (6,3) and (7,3) (see table 1.2). Even though the correlation is higher in those attributes, the values are still unimportant (lower than 0.8) in order to consider a strong correlation between them.

| Attributes | Range | Type |
|---|---|---|
| Fixed acidity | [3 ; 15.9] | Continuous |
| Volatile acidity | [0.1 ; 1.58] | Continuous |
| Citric acid | [0 ; 1] | Continuous |
| Residual sugar | [0.7 ; 23.5] | Continuous |
| Citric acid | [0 ; 1] | Continuous |
| Chlorides | [0.013 ; 0.611] | Continuous |
| Free sulfur dioxide | [2 ; 289] | Continuous |
| Total sulfur dioxide | [7 ; 440] | Continuous |
| Density | [0.9874 ; 1.0032] | Continuous |
| pH | [2.79 ; 3.9 ] | Continuous |
| Sulphates | [0.25 ; 1.36] | Continuous |
| Alcohol | [8 ; 14.9] | Continuous |
| Total sulfur dioxide | [7,440] | Continuous |

Table 1.1: Range and types of the attributes

| Attributes (by pairs) | Tag | Correlations |
|---|---|---|
| (7,10) | (density, alcohol) | -0.692 |
| (6,5) | (total sulfur dioxide, free sulfur dioxide) | 0.73 |
| (6,3) | (total sulfur dioxide, residual sugar) | 0.52 |
| (7,3) | (density, residual sugar) | 0.526 |

Table 1.2: Most correlated attributes

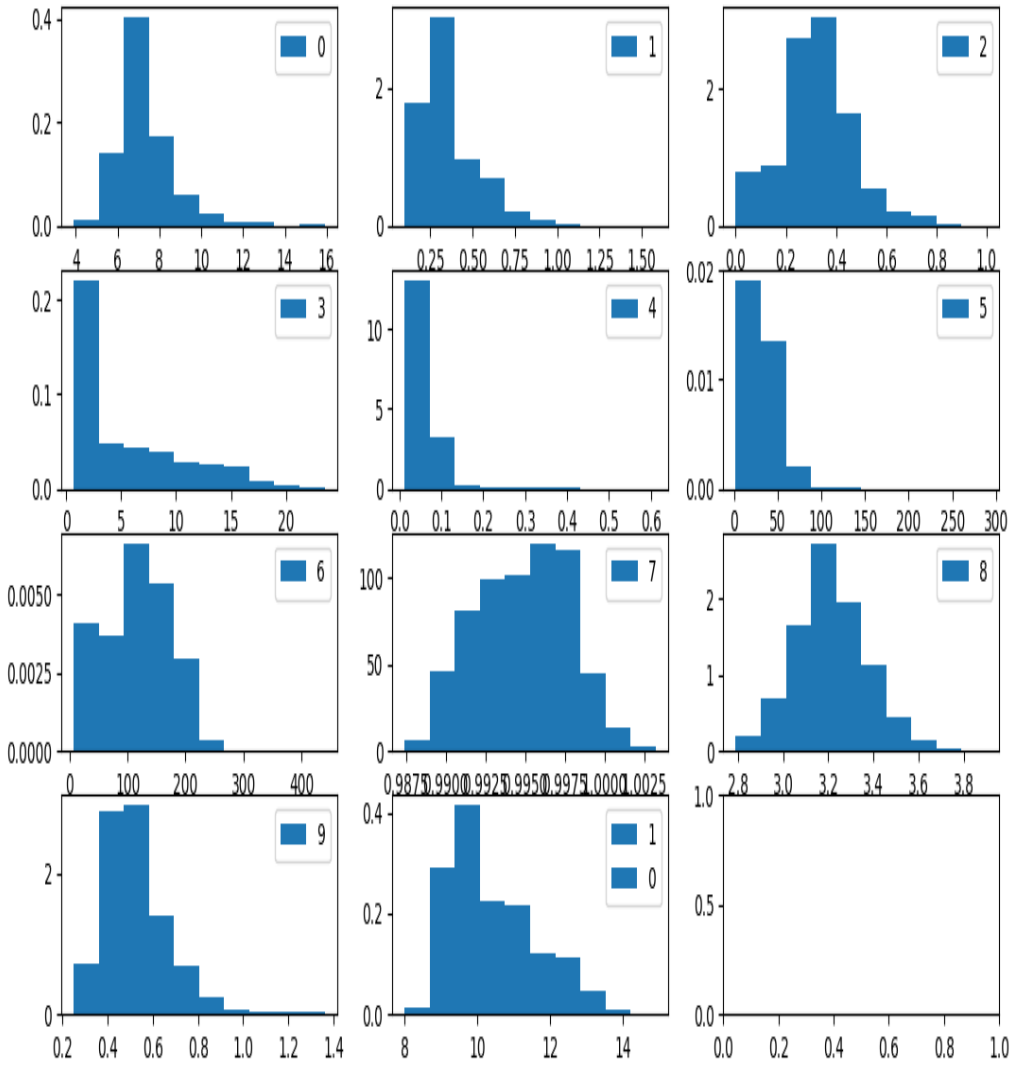


Figure 1.1: Histograms of the attributes (numerated) without modification

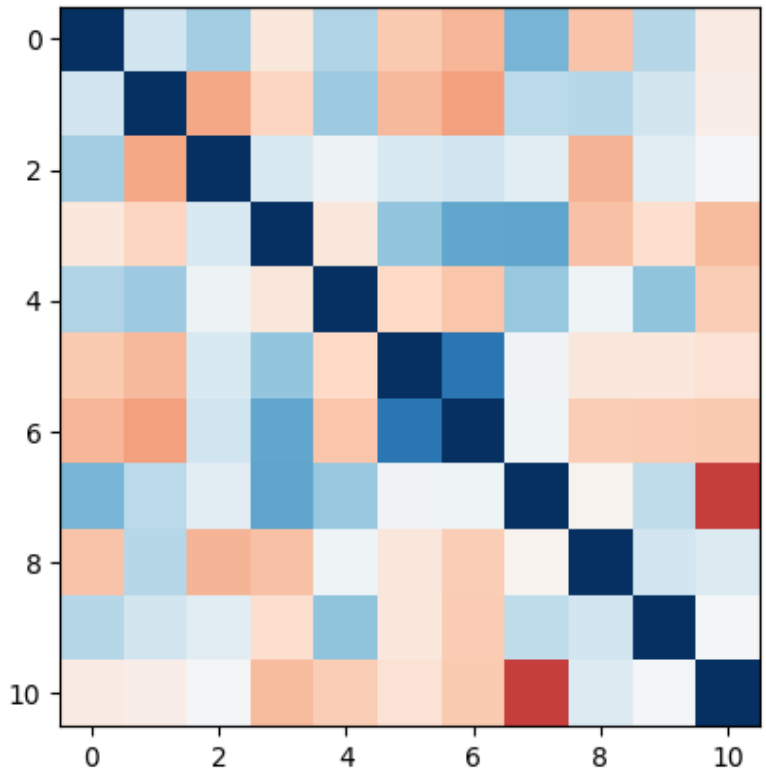


Figure 1.2: Correlations of the raw data (Pearson's coefficient)

### 1.1.2 PCA - Principal Component Analysis

Principal Component Analysis or PCA is a widely used technique for dimensionality reduction of the large data set. Reducing the number of components or features costs some accuracy and on the other hand, it makes the large data set simpler, easy to explore and visualize. Also, it reduces the computational complexity of the model which makes machine learning algorithms run faster.

**Steps Involved in PCA**

1. Standardize the data (with mean =0 and variance = 1).

2. Compute the Covariance matrix of dimensions.

3. Obtain the Eigenvectors and Eigenvalues from the covariance matrix

4. Sort eigenvalues in descending order and choose the top k Eigenvectors that correspond to the k largest eigenvalues (k will become the number of dimensions of the new feature subspace $k \leq d$, d being the number of original dimensions).

5. Construct the projection matrix W from the selected k Eigenvectors.

6. Transform the original data set X via W to obtain the new k-dimensional feature subspace Y.

Analyzing the eigenvalues obtained, we extracct from figure 1.3 how almost all of the information can be obtained using 4 or 5 dimensions. By drawing the cumulative graph of the eigenvalues, the same conclusion can be extracted. Hence, PCA could be implemented and see if the accuracy improves significantly.

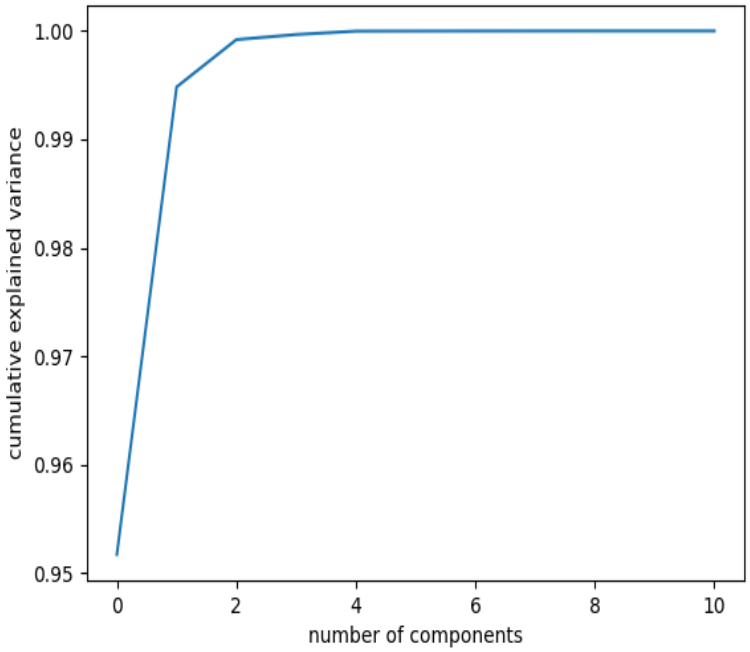


Figure 1.3: Cumulative explained variance according to the number of components after PCA

## 1.2 Classifiers

In order to perform the classification, we have performedd cross validation on the training set with k = 8. The decision for that value of k was taken due to one major factor: time. In fact, the size of the sample being 1839, the leave-one-out method was discarded due to time needed to perform it. Moreover, while performing k-fold with some classifiers, k = 8 appeared as an affordable value concerning time consumption, whilst performing a decent classification (> 80%). Hence every time the k-fold is performed, we are using k = 8 unless stated otherwise.

### 1.2.1 Generative models

In order to classify the data, different genereative models have been taken into account: mutlivariate gaussian classifier, tied covariance, Naive Bayes through a k-fold approach including a PCA approach. The results are shown in the table 1.3.

The precision of the models seems to be similar, however the Multivariate model has a greater precision so far with 81.4 % with PCA, and Naive Bayes performs the worst with 76 % (without PCA). The explanation of the sligh worse performance of Naive Bayes can be due to

the fact that when we perform the diagonalization of the covariances, we do so by putting zero in the other elements of the matrix, hence inevitably losing information.

So far, from the table it can be extracted that the MVG classifier with PCA is so far the best solution with a 81.4 % accuracy.

| Classifier | Accuracy (%) | Accuracy with PCA (%) |
|---|---|---|
| Multivariate | 80.3 | 81.4 |
| Tied Covariance | 79 | 79.7 |
| Naive Bayes | 76 | 79.7 |

Table 1.3: Different types of classifiers and their accuracy

### 1.2.2 Logistic regression

For the logistic regression model we used a k-fold approach (k =8), where in each one we changed the value of lambda. The values of lambda tested where: [1e-05, 0.0001, 0.0002, 0.0004, 0.0006, 0.001, 0.004, 0.01]. This was done in order to try to get the best model possible. This approach was performed with PCA too. The results are shown in table 1.4.

As the results in the table 1.4, the linear regression without PCA, performed overall better than the model with the PCA. The best performance of the PCA model was with lambda = 0.0004 resulting in an error rate of 16.6. The model without PCA, outperforms the model with PCA, with its best error rate being 14.4% (see value in blue).

| lambda | PCA error rate (%) | no PCA error rate(%) |
|---|---|---|
| 1.e-5 | 18.34 | 18.8 |
| 0.0001 | 19.21 | 17.5 |
| 0.0002 | 20.1 | 14.8 |
| 0.0004 | 16.6 | 14.4 |
| 0.006 | 24.9 | 21 |
| 0.001 | 22.3 | 18.3 |
| 0.004 | 20.5 | 20.5 |
| 0.01 | 18.8 | 15.7 |

Table 1.4: Error rates in (%) for linear regression with (and without) PCA

### 1.2.3 Confusion matrix and Optimal Bayes

Concerning the confusion matrix classification model, it has also been performed using PCA and without PCA, each done with a k-fold = 8. We have also take into account a prior of 0.5 for each class. By trying other priors the precision diminished, therefore we decided to stick with a prior of 0.5. Additionally, we decided to put a cost for a false positive of 2 (CFP = 2), while a cost for a false negative (CFN = 1). This decision was taken as we considered that missclassifying a "bad" quality wine is worse than missclassifying a good quality wine.

The results of computing the optimal Bayes decisions for the parameters previously chosen, are in the table 1.5. As it can be extracted from the table, using PCA results in a worse performance the raw data inn every classifier except for the Naive Bayes (with PCA 80.8 % of precision versus 77.91 %). Finally, the best model seems to be MVG with a precision of 85.5% and a DCF normalized of 0.432 (which is the best among the 3 models), which means that implementing the model has a positive impact in the classification.

| Classifier | Tied | MVG | Naive |
|---|---|---|---|
| Precision (%) | 81.86 | 85.5 | 77.91 |
| Precision (PCA) | 79.2 | 80.55 | 80.8 |
| DCF | 0.274 | 0.216 | 0.286 |
| DCF (PCA) | 0.312 | 0.291 | 0.33 |
| DCF normalized | 0.548 | 0.432 | 0.572 |
| DCF normalized (PCA) | 0.624 | 0.582 | 0.662 |

Table 1.5: Precision in (%) for different logistic regression models

### 1.2.4 SVM

We have performed the linear SVM with raw data and with PCA. The results with PCA underperform the ones with raw data: the best error rate with PCA being 17.9 % and without PCA 14.4 % (check results in tables 1.6 and 1.7 respectively).

However, the duality gap for the 17.9 % error rate for PCA data is 119.13. This value stands out among the others as really elevated . This could make us think that the reliability of the error rate obtained for c = 0.5 can be questioned.

On the other side, the raw data performed well for c = 0.04 and c = 0.05 with both error rates being equal to 14.4 % while the duality gap being 0 (check table 1.7 for results).

After performing linear SVM and having some satisifying results with the classification, we decided not to proceed with the kernel SVM due to the high computational time that took to perform the linear SVM.

| c | error rate | duality gap | primal |
|---|---|---|---|
| 0.001 | 29.3 | 0.0 | 1.122 |
| 0.005 | 24.5 | 0.0 | 5.938 |
| 0.01 | 26.2 | 0.0884 | 13.15 |
| 0.05 | 30.1 | 43.1 | 110.63 |
| 0.08 | 25.3 | 6.11 | 119.76 |
| 0.1 | 23.14 | 3.78 | 157.19 |
| 0.5 | 17.9 | 119.13 | 859.1 |
| 0.9 | 27.1 | 342.73 | 1057.13 |

Table 1.6: Error rate in (%) for different parameters in linear SVM with PCA

| c | error Rate | duality gap |
|---|---|---|
| 0.001 | 29.7 | 0 |
| 0.004 | 20.5 | 0.0056 |
| 0.005 | 21.8 | 0 |
| 0.006 | 20.1 | 0 |
| 0.007 | 16.6 | 0 |
| 0.008 | 27.5 | 0.0013 |
| 0.009 | 20.5 | 0.0001 |
| 0.01 | 17.5 | 0 |
| 0.02 | 19.2 | 0 |
| 0.03 | 17.9 | 0.0295 |
| 0.04 | 14.4 | 0.014 |
| 0.05 | 14.4 | 0 |
| 0.06 | 21 | 5.095 |
| 0.08 | 24.9 | 1.645 |
| 0.1 | 19.2 | 0.586 |
| 0.5 | 31.9 | 2259.61 |
| 0.9 | 18.8 | 97 |

Table 1.7: Error rate in (%) for different parameters in linear SVM without PCA

### 1.2.5 Gaussian mixture models

For this classsifiers we decided to choose as parameters alpha = 0.1, minimum value for the eigen vector 0.1. Moreover, we tested until gmm = 16 (1,2,4,8 and 16).

The choice for this parameters was based on the laboratory 10. Once the results appeared, we didn't change the parameters due to the fact that the results were satisfying (see table 1.8) and that the computation for every case was too big to repeat. Hence, having decent precision results we estimated that the the recomputation was not worth the possibility of improvement.

The results shown in table 1.8, illustrate that there is a tendency when increasing the model's precision by augmenting the gmm (gmm =16 is the most precise model in every scenario), this could be due to overfitting as the error rate of this method outperforms other models so far (9% against around 15% for other models). On one hand, the most precise model is the full covariance classifier with PCA previously computed, resulting in a 9.31% error rate, versus 9.71 % without PCA. On the other hand, with a lower complexity (gmm = 8), the precision is still good enough for the full covariance without PCA (11.8 %).

| Classifier | gmm = 1 | gmm = 2 | gmm = 4 | gmm = 8 | gmm = 16 | PCA | k-fold |
|---|---|---|---|---|---|---|---|
| Full covariance | 16.4 | 15.7 | 15.4 | 11.8 | 9.71 | No | 8 |
| Full covariance | 17 | 15.8 | 15.8 | 12.9 | 9.31 | Yes | 4 |
| Tied Covariance | 16.4 | 16.4 | 16.4 | 16.4 | 14.9 | No | 8 |
| Tied Covariance | 18.6 | 18.6 | 19 | 18.4 | 17.6 | Yes | 8 |
| Diagonal covariance | 22 | 21.6 | 19.2 | 15.1 | 13.5 | No | 8 |
| Diagonal covariance | 20.3 | 21.3 | 21.3 | 16.8 | 16 | Yes | 8 |

Table 1.8: Error rates in (%) for different GMM model classification

## 1.3 Model choice and results

With all the previously models taken into account, basing ourselves solely in the classification performance, what appears to be best at classifying the data is the GMM model , specifically for gmm =16 with a full covariance and PCA applied.

However, with a punctual classification performance does not mean that the model will classify better the test data. In fact, with gmm = 16 it appears as the model is overfitting itself

to the training data.Therefore, considering only one performance doesn't appear to be the best decision. The table 1.9 takes the best classification result for every classifier.

| Classifier | error Rate (%) |
|---|---|
| GMM = 16 without PCA | 9.71 |
| GMM = 8 without PCA | 11.8 |
| GMM = 16 with PCA | 9.31 |
| Logistic regression (c = 0.0004 ) | 14.4 |
| MVG (with PCA) | 18.6 |
| Optimal Bayes (MVG) | 14.5 |
| Linear SVM (c = 0.04) | 14.4 |

Table 1.9: Error rates in (%) for different classification models

What can be extracted from table 1.9 is that an error rate of around 14 % seems to be normal for the classification. Moreover, we are discarding the GMM models with g =16 as it appears to be overfitting. The model with MVG and PCA is discarded too because of its high error rate (18.6%). Finally, eventhough the gmm for g =8 might be seen as overfitting, the GMM model overall performed better than the others, therefore for g = 8 this model would be a great middle point between a worse performance (15.4 % for gmm = 4) and overfitting.

**Model chosen: GMM with g= 8 and no PCA**

The resulting error for GMM = 8 and without PCA on the data was 14.87 %, which is in the range of error rate that we have achieved with the other models in the train data.