# **Wine quality detection report**

##### **D’Ascenzi Giulia s287720, De Girolamo Patrizio s292497**

Report for the “Machine Learning and Pattern Recognition” course.

Academic year 2020-2021

# **Introduction**

T

he dataset used for the purpose of this analysis is related to red and white variants of the Portuguese "Vinho Verde" wine and is taken from the UCI repository.

The dataset was collected to predict human wine taste preferences. As a matter of fact, it associates the physiochemical characteristics of the analyzed wine to sensory evaluations made by experts.

The original dataset consists of 10 classes (quality 1 to 10), but for this project the dataset has been binarized, collecting all wines with low quality (lower than 6) into class 0 and good quality (grater than 6) into class 0. Wines with quality 6 have been discarded to simplify the task.

The dataset contains both red and white wines that were originally separated, whereas in this analysis they have been merged.

## **Classes Balance**

The dataset has been divided into a training set and a test set.The training set contains 613 samples belonging to the "high quality" class and 1226 belonging to the "low quality" class. On the other hand, the evaluation set contains 664 samples of class "high quality" and 1158 samples of class "low quality". Therefore, the classes are partially balanced.

The following graphs show the comparison between the number of samples of the two classes in both training and test data set. It emerges that both the number of samples and the ratio between the two classes in the different datasets is similar.

|  |  |
| --- | --- |
|  |  |

## **Attributes Information**

The input variables have 11 continuous features based on physiochemical tests: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, alcohol.

The output variable is a discrete value representing the quality of the wine sample (0 low quality/1 high quality)

## **Preprocessing and feature analysis**

The features of the dataset refer to different types of variables and have therefore different measuring scales.

Thus, in order to compare similarities between features, the dataset has been z-normalized, through centering and scaling to unit variance the features.

The histograms below show the distributions of the training dataset features. Orange histograms refer to high quality wines, whereas blue histograms to low quality wines.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

The analysis of the training data reveals that most of the features have an irregular distribution.

Consequently, the classification (especially of Gaussian based methods) may produce sub-optimal results. We therefore further preprocessed our data by "Gaussianizing" the features.

The gaussianization process allows mapping the features values to ones whose empirical comulative distribution function is well approximated by a Gaussian c.d.f. To do so, the features have been mapped to a uniform distribution and then transformed through the inverse of Gaussian cumulative distribution function.

The histograms below show the distributions of the gaussianized features.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

A correlation analysis of the Gaussianized features shows that feature 5 and 6 are strongly correlated.

The heatmaps showing the Pearson correlation coefficient

can be found below.

From right to left: whole dataset, high quality features, low quality features.

|  |  |  |
| --- | --- | --- |
|  |  |  |

This suggests that the classification may benefit from using PCA to map data to 10 or 9 uncorrelated features to reduce the number of parameters to estimate.

# **Classification of wine quality features**

## **Methodologies used for validation**

In order to understand which model is most promising and to assess the effects of using PCA, both single fold-validation and K-Fold cross-validation have been adopted.

At first, a single fold approach has been used, since the training with this approach is faster. Indeed, single fold approach consists in splitting the training dataset into two subsets, one (66% of the original set) for development and the other one for validation. Subsequently, the K-Fold approach has been used to get more robust results. In this case, the training set are iteratively split into 5 folds, 4 used for training and 1 for validation, after being shuffled. The validation scores have been eventually put together and used to compute the performance metrics. In this way, there is more data available for training and validation.

This document focuses on the analysis of the balanced uniform prior application:

(π, ,) = (0.5,1,1)

However, two other unbalanced applications have been considered:

(π, ,) = (0.9,1,1) and (π, ,) = (0.1,1,1)

In the fist part, the main aim of the analysis was to choose the most promising approach. Therefore, the performances have been measured in terms of normalized minimum detection costs, namely the costs that would be paid by making optimal decisions for the validation set through the use of recognizers scores.

## **MVG CLASSIFIERS**

The first type of classifier analysed are the multivariate gaussian classifiers that assume that the data, given the class, can be described by a Gaussian distribution.

The following table show the results in terms of minimum DCF of the gaussian classifiers studied.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Single Fold** | | | **5-Fold** | | |
|  |  |  |  |  |  |  |
| **Raw Features – no PCA** | | | | | | |
| **Full - Cov** | 0.304 | 0.812 | 0.777 | 0.312 | 0.842 | 0.778 |
| **Diag -Cov** | 0.437 | 0.875 | 0.818 | 0.420 | 0.921 | 0.845 |
| **Tied Full-Cov** | 0.334 | 0.733 | 0.779 | 0.333 | 0.748 | 0.812 |
| **Tied Diag- Cov** | 0.412 | 0.901 | 0.832 | 0.402 | 0.932 | 0.866 |
| **Gaussianized Features – no PCA** | | | | | | |
| **Full - Cov** | 0.270 | 0.740 | 0.807 | 0.306 | 0.790 | 0.784 |
| **Diag -Cov** | 0.456 | 0.848 | 0.860 | 0.448 | 0.914 | 0.834 |
| **Tied Full-Cov** | 0.348 | 0.812 | 0.867 | 0.354 | 0.884 | 0.803 |
| **Tied Diag- Cov** | 0.451 | 0.891 | 0.839 | 0.451 | 0.942 | 0.879 |
| **Gaussianized Features – PCA (m=10)** | | | | | | |
| **Full - Cov** | 0.312 | 0.817 | 0.779 | 0.328 | 0.862 | 0.807 |
| **Diag -Cov** | 0.362 | 0.747 | 0.780 | 0.378 | 0.824 | 0.804 |
| **Tied Full-Cov** | 0.324 | 0.682 | 0.791 | 0.328 | 0.753 | 0.809 |
| **Tied Diag- Cov** | 0.335 | 0.656 | 0.822 | 0.334 | 0.782 | 0.822 |
| **Gaussianized Features – PCA (m=9)** | | | | | | |
| **Full - Cov** | 0.300 | 0.817 | 0.783 | 0.319 | 0.814 | 0.800 |
| **Diag -Cov** | 0.349 | 0.764 | 0.810 | 0.368 | 0.814 | 0.804 |
| **Tied Full-Cov** | 0.331 | 0.677 | 0.786 | 0.327 | 0.752 | 0.816 |
| **Tied Diag- Cov** | 0.337 | 0.673 | 0.822 | 0.335 | 0.783 | 0.825 |

The diagonal covariance models (both full diagonal and tied diagonal) do not give good results if compared to the other models with or without Gaussianization and PCA. These models work under the naive-bayes-assumption, according to which the different components for each class are uncorrelated. Therefore, in this case, this assumption does not produce accurate results. Applying PCA slightly improves the performance, probabily because by removing the low variances directions the whithin-class correlation decreases

The tied full covariance model performs pretty good on the validation data, confirming the similiarity between classes shown by the correlation analysis. It performs better than the diagonal models accounting the within-class correlations. In this case, PCA does not help in producing better results.

The best-performing model is the full covariance model, which is able to account for correlations. What is more, having enough data compared to the dimensionality of the samples, the results we succeed obtaining robust results.

Gaussianization fails in improving significantly the results, although it still performs slightly better if compared to raw features.

Even if it does not help increasing much the performances, PCA can still be used to reduce the number of parameters and

therefore to reduce the complexity of the model. Nevertheless, given the limited effectiveness, it will not be considered for the future analysis.

The results between single-fold and K-fold are consistent, suggesting that the amount of data is enough for validation and model training also when it comes to the single-fold set up.

None of the models produce accurate results for the unbalanced applications.

In summary, the best candidate is currently the MVG model with full covariance matrices. The chosen one is the one with Gaussianized features, since it shows slightly better results than the one with raw features, and the K-fold approach, even if it is a bit worse than the single fold result. Nonetheless, this approach provides more robust results.

Since the best-performing models are the full covariance and the tied full models (which respectively have a quadratic surface rule and a linear surface rule), the decision was to proceed by analyzing both the quadratic and the linear models.