

# Features Extraction & Red-Wine Quality Predictions

Università della Calabria

Dipartimento di Ingegneria Informatica , Modellistica , Elettronica e Sistemistica  
(DIMES)

Nicola Corea

**Abstract.** The aim of this work is to present a new solution to the problem of red wine quality prediction. Several articles provides different solutions to the above problem , based on single classifiers or ensemble techniques, whose results in term of accuracy on test data are lower than 70 %. As we will see , after appropriate pre-elaborations of the data , the accuracy of the classifiers can be increased up to 20-30% respect to the current solutions. The application of techniques for dimensionality reduction , like selection and extraction , can be used to reduce the number of features taken in consideration , and loosing at the same time as little as possible on the accuracy. The dimensionality reduction allow us to obtain model simpler than the previous classifiers ,and reduces the computational cost required to train them. We will conclude our discussion by training a multi-layer perceptron (MLP) for classification purposes, and we are going to compare the results with those obtained by simple and standard classifiers.

## I. Introduction

The aim of this article is to train different classifiers with the purpose to predict the red wine quality on the base of the wine's characteristics. To train our classifiers we refer to the dataset available from UCI machine learning repository. The dataset , thanks to the work of P. Cortez , A. Cerdeira , F. Almedia , contains 1599 examples of the red portuguese "Vinho Verde" wine , classified on base of different characteristics , such as : 'fixed acidity' , 'volatile acidity' , 'citric acid' etc... . On the base of these features , the quality of the wine is represented by an integer in the closed interval [3,8]. The value '3' will be refered to a low quality of the wine , instead , the integer '8' will represent an high quality. The performance of the classifiers will be evaluated in terms of accuracy. To this end , with reference to the confusion matrix , represented in **Figure 1.1** , said

$$ERR = \frac{FP + FN}{FP + FN + TP + TN}$$

the classification error , the accuracy of the classifier is given by

$$ACC = 1 - ERR = \frac{TP + TN}{FP + FN + TP + TN}$$

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

**Figure 1.1** Confusion Matrix

We start our discussion , by presenting the results obtained by the classifiers without the modification (pre-processing) of the labels associated to the data. In particular , according to [1] , we will see that the accuracy on the test data , is lower than 70%. The aim of this work , is to find possible solutions with the intention to improve the performance of the classifiers. As we will see , with an appropriate pre-processing phase on the class labels , the accuracy of the predictors can be bring to 95-96% on test data. After doing so , with the purpose to use ensemble techniques such as Boosting , Bagging etc.. , we present different solution to dimensionality reduction , like SBS (Sequential Backward Selection) and PCA (Principal Components Analysis). With the application of these techniques we will obtain simplex classifiers , and of course , we have a reduction on the

computational cost required to train our classifiers. The article end with the training of a multi-layer perceptron (MLP) for classification purposes , and we analyze and compare the results with those obtained in the previous section.

## II. Problem Analysis

With the aim of evaluating the performance of the model obtained , before leaving it free to operate in the real world , we split the dataset in a 70% for training and a 30% for test. The next step, of considerable importance , is the standardization (or normalization) of the feauteres. Some algorithms , such as KNN (K-Nearest-Neighbors) or SVM (Support – Vector-Machine) are not invariant respect to the feature scales. For example , remember that in the KNN algorithm , the selection of the k-neighbors is equals to find the k examples in the dataset at minimum distance from the example that we want to classify. If we choose , as metric , the Euclidean Norm (Squared)

$$|x - x_k|_2^2 = \sum_{i=1}^n (x_i - x_{ki})^2$$

we can see how the distances are most influenzed by the greatest features. So , before proceeding , we have to standardize the features. In **Table 1.1** we report some aspects of the features , like means and variance , and the minimum and maximum values. As we can see, the values of some features , such as , ‘Fixed Acidity’ , ‘Residual Sugar’ , are greater than others.

Given means and variance for each characteristic , the examples will be tranformed in according to the following equation

$$x_{std}^i = \frac{x^i - \mu_x}{\sigma_x}$$

and they can be interpreted as realization of a normal distribution with 0 mean and 1 deviation. So now , we can train our classifiers , and we can evaluate the accuracy of the single predictors.

	"fixed acidity"	"volatile acidity"	"citric acid"	"residual sugar"
count	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806
std	1.741096	0.179060	0.194801	1.409928
min	4.600000	0.120000	0.000000	0.900000
25%	7.100000	0.390000	0.090000	1.900000
50%	7.900000	0.520000	0.260000	2.200000
75%	9.200000	0.640000	0.420000	2.600000
max	15.900000	1.580000	1.000000	15.500000

**Table 1.1** Characteristics Aspects

Let’s proceed with the training of the following classifiers:

- 1) KNN (K-Nearest-Neighbors)
- 2) SVM Kernel (RBF Kernel)
- 3) DecitionTree Classifier (Entropy)

Remember that we evaluate the performance of our classifiers on base of the accuracy. In the following table we report the results obtained both in training and testing phase

Classifier	Training	Test
SVM	0.94	0.69
KNN	0.66	0.58
DTC	0.67	0.59

**Table 1.2** Results

As we can see from the Table 1.2 , and as stated in [1] , the best classifier for this kind of problemi is the SVM algorithm. In particular, with reference to the following criterior

$$J(w) = \frac{1}{2} |w|_2^2 + C \left( \sum \xi_i \right)$$

for the selection of the hyperplane , and with respect to the following kernel function

$$k(x_i, x_j) = \exp(-\gamma |x_i - x_j|)$$

we have choosen the hyperparameters as  $C = 10.0$  ,  $\gamma = 0.25$ . We can try to improve the accuracy by using ensemble techniques. But in this dataset we have 13 features , so the number of the classifiers in the ensemble has not to be large , related to computational problems.

Ensemble	Training	Test
MV	0.79	0.63
RF	0.99	0.68
AB	0.56	0.54

**Table 1.2** Ensemble Results

The Table 1.2 shows the results obtained by using different ensemble techniques like Boosting , Bagging and HardVoting (Majority Voting). In particular in the majority ensemble we have used the previous trained classifiers.