Exploratory Analysis for Wine Classification by Type of Wine Cultivation

Diogo Eduardo Lima Alves

***Resumo*— Este artigo implementa algoritmos clássicos para o propósito de classificação. A base de dados utilizada é pública e composta pelos atributos químicos de diferentes vinhos. Há três diferentes classificações que correspondem a três diferentes tipos de cultivo.**

***Palavras-Chave*— vinho, classificação**

***Abstract*— This paper implements classical algorithms for classification purpose. The database we use is public and composed by many chemical attributes of different wines. We have three different classifications and each one corresponds to a different type of cultivation.**

***Keywords*— wine, classification**

I. Introduction

The database used was created in 1990 and it is composed by the attributes obtained in a chemical analysis of wines whose grapes were cultivated in the same region in Italy but using three different types of cultivation. This chemical analysis identified 13 attributes of each wine. The classification purpose is to classify a specific wine by cultivation type based on the attributes obtained in the chemical analysis. We use classical supervised learning algorithms to achieve our goals.

II. Background

The database is composed by the following items:

- (V1) Class.
- (V2) Alcohol.
- (V3) Malic acid.
- (V4) Ash.
- (V5) Alcalinity of ash.
- (V6) Magnesium.
- (V7) Total phenols.
- (V8) Flavanoids.
- (V9) Nonflavanoid phenols.
- (V10) Proanthocyanins.
- (V11) Color intensity.
- (V12) Hue.
- (V13) OD280/OD315 of diluted wines.
- (V14) Proline.

We have 178 different instances in the database and any missing value. The attributes can be integer or real. See complete details in Figure 1.

This paper was partially funded by CAPES

Data Set Characteristics:	Multivariate	**Number of Instances:**	178	**Area:**	Physical
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	13	**Date Donated**	1991-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	1209101

Fig. 1. Data Characteristics.

The database used is part of UC Irvine Machine Learning Repository [2]. Following we briefly describe the working process of each algorithm.

Decision Tree: It creates a rank of attributes based on the information gain and divide the instances based on the attribute on the top of the rank. This process is done multiple times until we get a good classification. Each node acts as a "test" on an attribute and each branch represents the outcome of the test and each leaf node represents a class label.

Dendrogram: Suppose we have five features. The hierarchical clustering dendrogram would show a column of five nodes representing the initial data and the remaining nodes represent the clusters which the data belongs, with the arrows representing the dissimilarity. The distance between merged clusters increases with the level of the merger. The height of the dendrogram also determines how many clusters it has and can be analyzed to define the optimal number of clusters.

k-Nearest Neighbors: For each new instance we let the k closest neighbors vote to define the class of this instance. the k is typically chosen to be an odd number.

Naive Bayes: This classifier considers each feature to contribute independently to the probability that a specific instance belongs to a class, regardless of any possible correlations between features.

The database used [2] have been intensively studied for classification purposes. For other approaches or evaluation by comparison see [1] [3] [4] [5].

III. Proposal

First of all we analyze the database in order to acquire good attributes selection for each algorithm. Depending on algorithm used it is better to select few attributes or use all them. We can make this type of analysis using measures as mean, median, dispersion, correlation, etc.

We also preprocess the data, although the database we are working with has no missing values we normalize the attributes in order to acquire better results with algorithms that are sensible to denormalized data.

The approaches used are: Decision Tree, Dendrogram, k-Nearest Neighbors and Naive Bayes. We also construct measures of effectiveness for each approach, in general, we use accuracy rate based on confusion matrix.

IV. Results

A. Analyzing Data

In order to acquire knowledge about our data, we extract plots and metrics from all data or isolated features. Among them: correlation, histograms, boxplots, summaries (mean, median, etc.), dispersion, etc. However we only discuss the metrics and analysis which were decisive for decision making about selecting features or algorithms, in other words, the ones we are able to use in order to extract information.

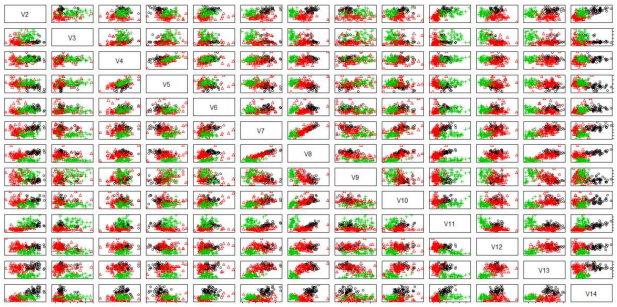

Fig. 2. Scatter Plot.

We first analyze the scatter plot of our attributes (see Figure 2). This analysis showed that comparing two by two the attributes V13 and V14 are the most interesting in order to be used in a simpler classification technique, a linear or distance based algorithm, for example (see Figure 3). Then, based on data dispersion, we explore k-NN algorithm later in this paper.

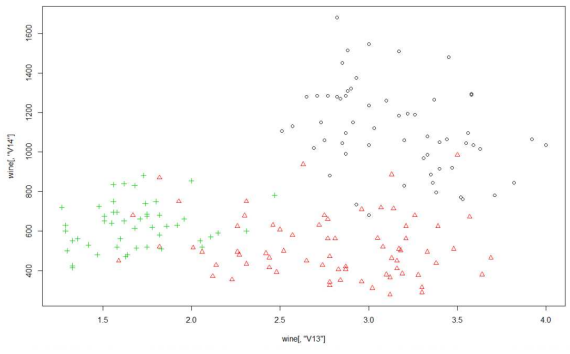

Fig. 3. V13 x V14 Scatter Plot.

We also analyzed the correlation and we identified that the pairs [V7,V8], [V7,V13], [V7,V14], [V8,V13], [V8,V14] and [V13,V14] are the ones with bigger linear correlation coefficients.

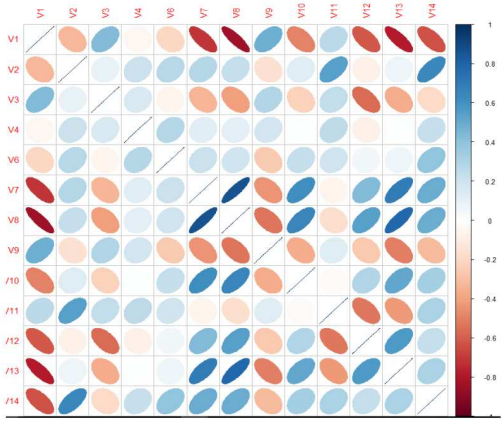

Fig. 4. Correlation.

B. Dendrogram

The Dendrogram is a hierarchical clustering approach, although it is not often used for classification problems as it is based on dissimilarity or a specific distance measure between elements we use it to analyze instances based on their dissimilarity and the possible efficiency of algorithms based on distance.

Using the dendrogram approach (see Figure 5) we realize the class 1 is the most separable by this algorithm but class 2 and 3 had no good results in separating data.

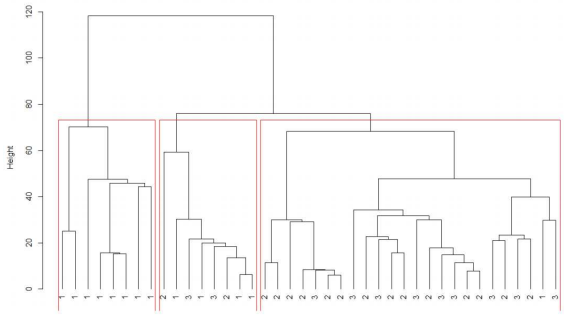

Fig. 5. Dendrogram.

Because of that we can guess algorithms as k-NN has reasonable efficiency to classify class 1 but the average accuracy is not expected to be the best one when compared with other algorithms.

C. Sampling Strategy

In the next algorithms we use sampling strategy to be able to measure the accuracy rate. We select 70% of instances to construct the training set and 30% to construct the validation set. We use the training set to construct the model and the validation set to measure our results obtained by the model. The sets are construct randomly which means in each execution the accuracy rate can be different based on the instances selected for each set.

In these algorithms we will perform many executions in order to evaluate the accuracy interval achieved by each algorithm. We use the validation set to construct a confusion matrix

which compares the real classes of instances in validation set with the classes predicted by the model generated. The accuracy rate is the coefficient obtained by the sum of the confusion matrix diagonal (TP+TN) divided by the sum of all entries in the confusion matrix (TP+FP+FN+TN).

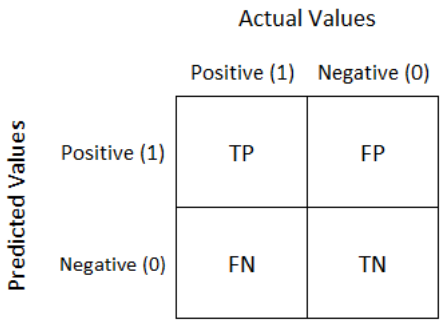

Fig. 6. Confusion Matrix

In Figure 6 we have the confusion matrix structure and its components:

- TP - True Positive.
- TN - True Negative.
- FN - False Negative.
- TN - True Negative.

These values are used to construct different measures as recall, precision, specificity, accuracy and AUC-ROC Curves. The confusion matrix also can be generalized for problems with more than 2 classes.

D. Decision tree

In Figure 7 we can see the rules hierarchy achieved by an example of decision tree generated in our experiments, recall decision trees are important even for analysis once the attributes on the top of the decision tree are the attributes with bigger information gain coefficients. Then we can identify the attribute V11 as the attribute with the bigger information gain coefficient in this specific execution.

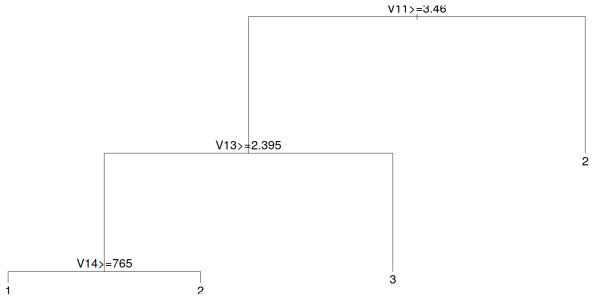

Fig. 7. Decision Tree.

We analyze the confusion matrices of the decision trees generated through the different executions and we acquired accuracy values between 0.74 and 0.95 depending on the execution. The accuracy obtained, in general, was bigger than 0.9 which is a really good value, considering decision tree is a simple approach and easy to understand.

E. k-NN Algorithm

By our dispersion analysis, we selected attributes V13 and V14 to be used in k-NN algorithm. The bigger accuracy rate was 0.8723 with $k = 3$ and the complete measures, including sensitivity, specificity, etc, can be seen in Figure 8.

```
Overall Statistics

               Accuracy : 0.8723
                 95% CI : (0.7426, 0.9517)
    No Information Rate : 0.4681
    P-Value [Acc > NIR] : 8.434e-09

                  Kappa : 0.7973

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3
Sensitivity            1.0000   0.9091   0.6667
Specificity            0.9706   0.8400   0.9714
Pos Pred Value         0.9286   0.8333   0.8889
Neg Pred Value         1.0000   0.9130   0.8947
Prevalence             0.2766   0.4681   0.2553
Detection Rate         0.2766   0.4255   0.1702
Detection Prevalence   0.2979   0.5106   0.1915
Balanced Accuracy      0.9853   0.8745   0.8190
```

Fig. 8. k-NN Statistics.

In [3] the authors achieved an 0.94 accuracy coefficient using k-NN algorithm with $k = 3$ and modified approach using other techniques. Then our simpler implementation of k-NN algorithm achieved a reasonable though inferior result.

F. Naive Bayes

Finally we use Naive Bayes algorithm. We use all the 13 features as input. The accuracy obtained by Naive Bayes is surprisingly excellent, in some executions it had a perfect accuracy (accuracy $= 1.0$) and never had an accuracy less than 0.9 in any of our executions.

TABELA I
ACCURACY TABLE.

	k-NN Algorithm	Decision Tree	Naive Bayes
Bigger Accuracy	0.8723	0.95	1
Average Accuracy	0.75	0.9	0.98

By Table I we can see Naive Bayes was the algorithm with bigger accuracy coefficient considering the bigger accuracy presented and the average one.

V. CONCLUSION

This paper analyzes different approaches and algorithms results implemented to classify the wine database [2]. After the analysis we identified Decision Tree and Naive Bayes with the bigger accuracy coefficients.

Although Naive Bayes is, in general, an approach to be considered when we want rapid implementation and reasonable results are sufficient, we can see that in this problem it was an excellent and surprising results approach, justifying its use and exploitation in other scenarios.

The Decision Tree is also a rapid implementation algorithm and because of that is very used to identify the most significant

features and features correlation, but it presents a really good accuracy level in our problem.

Depending on the problem and the purposes the accuracy obtained by k-NN can be useful, but considering the results achieved by simpler approaches it is not a good result.

Despite recent advanced techniques we can realize the effectiveness of classical algorithms in classifying real world data. It is clear that different problems are better treated by different approaches, but it can be a good idea to implement classical and already intensively studied algorithms because they may solve problems with few effort.

REFERÊNCIAS

[1] Jennifer G Dy and Carla E Brodley, *Feature selection for unsupervised learning*, Journal of Machine Learning Research **5** (2004), 845.

[2] M. Forina, *Wine data set*, 1990.

[3] Yuan Jiang and Zhi-Hua Zhou, *Editing training data for knn classifiers with neural network ensemble*, ISNN (1), 2004, p. 356.

[4] Stefan Mutter, Mark Hall, and Eibe Frank, *Using classification to evaluate the output of confidence-based association rule mining*, Australian Conference on Artificial Intelligence, 2004, p. 538.

[5] Ping Zhong and Masao Fukushima, *A regularized nonsmooth newton method for multi-class support vector machines*, Mar, 1.