Introduction to Machine Learning: Leaf identification

Silvia Imeneo[1], Piero Pettenà[2], and Tommaso Tarchi[3]

[1,2,3]The three authors equally collaborated in the implementation of each step of the study and in the drafting of this final report.

Course of AA 2022-2023 - Data Science and Scientific Computing

1 Problem statement

1.1 Data description

The goal of this project is to propose a suitable method to identify the plant species of a given leaf starting from the observation of several shape and texture attributes of the leaf itself.

We used as a basis the "leaf dataset" [1], which gathers 340 observations of different leaves. The dataset contains 14 numerical variables providing details about several characteristics of each leaf, and one categorical variable which is the class species the observed leaf belongs to. The numerical variables constitute the input of our ML system, while the categorical one is the output. We are in the case of a multiclass classification problem, with slightly unbalanced classes.

Our readers are encouraged to consult the publicly available referenced text should they look for more detailed information about the dataset used.

1.2 Assessment and performance indexes

We decided to assess our learning techniques with respect to their efficiency, in terms of computational time needed, and effectiveness. Since we have slightly unbalanced classes, we used the weighted accuracy as effectiveness index. Having few observations, and very few specimens for some classes, we decided not to split the data into static training/testing sets in order not to lose relevant information during the learning phase. Instead, we assessed the techniques by estimating the weighted accuracy on a 8-fold cross validation performed after having shuffled the data.

2 Design phase

2.1 Pre-processing

Out of the 40 different plant species classified during the study, only 30 are provided in the dataset and they are those referring to plants exhibiting simple leaves. Data on plants exhibiting complex leaves are not present and this is why they don't constitute a possible output.

Before using the dataset for our analysis, we removed the variable indicating the specimen number of each observation (column 2), since we did not consider it to provide useful information for the prediction of the leaf species. All remaining variables are used to build the machine learning system.

2.2 Defining implementation steps

The presence in the "leaf dataset" of both observations and responses, made us opt for a supervised learning tool instead of an unsupervised one.

We decided to compare five learning techniques: the single decision tree, the Random Forest, the Support Vector Machine (adapted for multiclass classification), the Naive Bayes, and the k-Nearest Neighbors. The steps that we followed are: looking for the optimal hyperparameters for each learning technique using grid search; learning the most effective model for each technique; assessing each model and choosing the best one to solve our problem.

In the hyperparameter tuning phase, we decided to use the weighted accuracy as index of effectiveness and the 8-fold cross validation as learning/testing division method.

3 Implementation

As anticipated, we started with the hyperparameter tuning step.

- For the single tree: we compared the Gini Index with the cross-entropy, and we looked for the optimal n_{min} over a range from 1 to 20.
- For the Random Forest: we tested the values 100, 200, 500, 700 and 900 for the parameter n_{tree}; we used the default and reasonable value $\sqrt{14}$, rounded to 4, for the parameter n_{vars}; and we compared the Gini Index with the cross-entropy.
- For the SVM: we applied the grid search together with the data standardization step; we looked for a proper value for c in a 22-value logarithmic range from -10 to 11; we compared Linear, Polynomial, Gaussian and Sigmoid kernels; we tested d over a range from 2 to 5, tested γ in a 13-value logarithmic range from -9 to 3; and we compared OVO with OVA for the application of SVM to our multiclass problem.

- For the Naive Bayes: no grid search was needed, since this technique has no hyperparameters. We assumed the likelihood of the features to be Gaussian.
- For the kNN: we looked for the optimal k over a range from 1 to $340 - \lfloor 340/k \rfloor$; and we compared the following distance functions: cosine, Euclidean and Manhattan.

The optimal hyperparameters that we obtained from the grid search are those shown in Table 1.

Learning technique	**Optimal hyperparameters**
Single tree	Cross-entropy, $n_{min} = 4$
Random Forest	Gini, $n_{tree} = 200$
SVM	Gaussian kernel, $\gamma = 0.0001$, $c = 100000$, OVO
kNN	$k = 5$, Manhattan

Table 1: Optimal hyperparameters for each learning technique.

4 Assessment

With the hyperparameters found, we fitted the optimal model for each learning technique and then proceeded to the assessment and comparison steps.

4.1 Efficiency

To assess the efficiency, we considered the time (in minutes) needed by each learning technique for both the hyperparameter tuning and the model fitting phases. The times we measured are indicated in Table 2. SVM resulted to be the most time consuming technique due to the long time needed by the hyperparameter tuning phase. We were expecting this to happen, due to the amount of parameters that we decided to test and their ranges. Regarding the model fitting phase, the Random Forest was the slowest one.

Nevertheless, for no learning technique the overall time needed was too high for our constraints, nor significantly high in general, so efficiency had a small influence on our final decision about the problem solution.

	Tree	**RF**	**SVM**	**NB**	**kNN**
Grid Searching	0:05	1:11	5:29	0:00	0:16
Model fitting	0:02	0:26	0:03	0:01	0:01
Total time	0:07	2:37	5:32	0:01	0:17

Table 2: Time in minutes required by each learning technique.

4.2 Effectiveness

As anticipated, we used the weighted accuracy as effectiveness index. The result that we observed indicated the SVM as the most effective technique, with an average effectiveness rate of 0.79, closely followed by the Random Forest, with a rate of 0.78. All the other techniques presented a lower effectiveness, as shown in Figure 1 below.

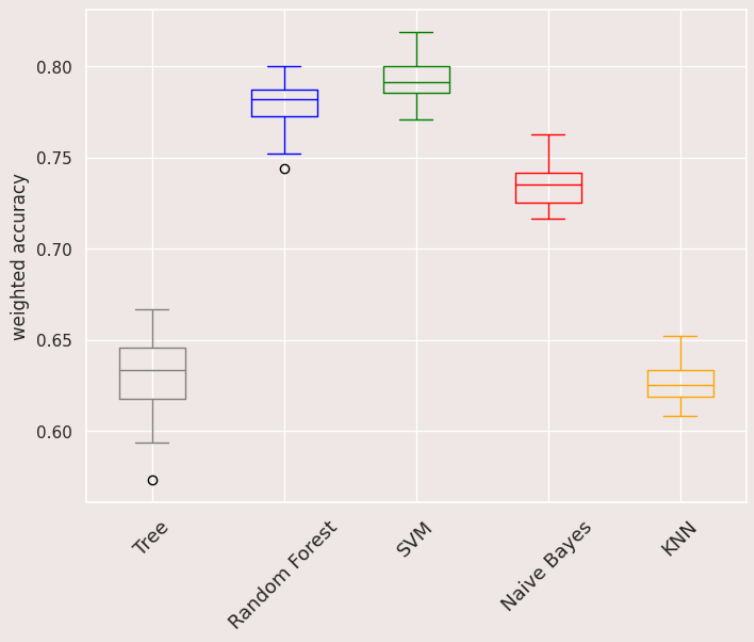

Figure 1: Boxplot of weighted accuracy

5 Results

The analysis of the results of both efficiency and effectiveness led us to elect the SVM as the most suitable technique to solve our problem.

However, it is worth noting that, as shown in the boxplot above, the difference in effectiveness between SVM and Random Forest is not that clear. Furthermore, the dataset used was not very large: it is likely that a different sample extracted from the same population would have given different results, maybe suggesting the choice of Random Forest instead of SVM.

References

[1] Rubim Almeida da Silva Pedro F. B. Silva Andre R. S. Marcal. *"leaf" dataset*. `https://archive.ics.uci.edu/ml/machine-learning-databases/00288/`. Accessed: 2023-01-09.