Introduction to Machine Learning and Evolutionary Robotics project: Leaf Identification

Simone Brusatin[1]

[1] problem statement, solution design, solution development, data gathering, writing

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

1 Problem Statement and Dataset Description

The goal of the project is to design, develop and present a solution, based on machine learning techniques, to the leaf classification problem, that is "given a leaf, determine its species". Note that we will be considering the leaf as a numerical vector. The pre-processing phase, in which a physical leaf is transformed into a vector, has already been done, so it is not part of our machine learning system.

The `leaf` dataset used is publicly available at [3]. The dataset consist of 340 rows and 16 columns, without missing values. We will consider the column "Species" as the response $y \in Y$, while, among the other 15 columns we will drop the "Specimen Number" column as it carries no useful information for prediction. The remaining 14 columns consist of numerical features $x_j \in X_j \subseteq \mathbb{R}, j \in \{1, \dots, 14\}$ that describe the shape and texture of the leaves. The size of the problem is $n \times p = 340 \times 14$, where n is the number of observations and p is the number of features. More detailed information on the dataset and the features can be found in the `Readme.pdf` file available with the dataset or at [4].

The response set Y is categorical, because it has no intrinsic ordering, with 30 possible values. The dataset is well-balanced, with number of observations per species ranging from 8 to 16.

The problem consists in correctly identifying the species of different leaves, among 30 possibilities, based on the 14 numerical features.

Formally:

$$f : X \subseteq \mathbb{R}^{14} \to Y$$

2 Assessment and Performance Indexes

For assessing the performance of the solution(s) we use both accuracy and weighted accuracy, with 10 fold cross validation. Note that since the dataset is well balanced, the two metrics will have similar values but nonetheless we will prefer weighted accuracy for choosing the best model. We will also report the standard deviation of the performance indexes.

3 Approach to the Problem

Our proposed approach is to try different supervised machine learning techniques. Namely we will use: decision tree, random forest, SVM with 3 different kernels (linear, polynomial and gaussian) and k nearest neighbours.
For each learning technique we will perform hyperparameter tuning through grid search with 10CV, considering weighted accuracy. Once the best parameters for each technique are found, we will perform an all-out comparison between the models and choose the most appropriate one.

4 Experimental Evaluation

4.1 Hyperparameter Tuning

In this subsection we will report the grids used for hyperparamter tuning. Note that all SVM models employ standard scaling and support multiclass classification through the *one-vs-one* technique (for more details see [2]).

Tree For the parameter n_{min} (minimum number of samples per node) we considered the values $\{1, 2, 3, 4, 5\}$ and two different node impurity measures: *Gini index* and *Cross entropy*.

Random Forest Although the technique has two parameters, n_{trees}, n_{vars}, it can be considered hyperparameter free as there are good default values: $n_{trees} = 500, n_{vars} = \lceil \sqrt{p} \rceil = 4$.

Linear SVM The only parameter is the weighting parameter c, searched among $\{0.1, 0.5, 1, 5, 10, 50, 100, 250, 500, 750, 1000\}$.

Polynomial SVM Other than the weighting parameter $c \in \{0.1, 0.5, 1, 5, 10, 50, 100, 250, 500, 750, 1000\}$, we also have the degree of the kernel $d \in \{2, 3, 4, 5, 6, 7, 8\}$.

Gaussian SVM We have the γ parameter chosen among $\{0.01, 0.001, 0.0001\}$ and the weighting parameter $c \in \{0.1, 0.5, 1, 5, 10, 50, 100, 250, 500, 750, 1000\}$

kNN We have two parameters: the number of neighbours $k \in \{1, 2, \ldots, 30\}$ and the distace d, chosen between the *euclidean distance* and the *manhattan distance*.

4.2 Chosen parameters

In the table below we report the combinations of parameters which correspond to the highest weighted accuracy for each learning technique, based on a 10 fold CV.

Tree	$n_{min} = 1$	impurity measure = *Gini index*
Random Forest	$n_{trees} = 500$	$n_{vars} = \lceil \sqrt{p} \rceil = 4$
Linear SVM	$c = 5$	
Polynomial SVM	$c = 50$	$d = 2$
Gaussian SVM	$c = 1000$	$\gamma = 0.01$
kNN	$k = 1$	$d = manhattan$

4.3 Results

In this section, we fit the models with the hyperparameters seen in the previous subsection. We measure and compare mean and standard deviation of accuracy and weighted accuracy, obtained using 10 fold cross validation.
In figure 1 below, we can see that the SVM with the linear kernel outperforms the other models with a weighted accuracy of 86.08% and an accuracy of 84.71%. Its standard deviation is considerably high, being 6.29% for w. accuracy, but we still consider it better than the other models. Random Forest has the smallest variance, for both accuracy and weighed accuracy. The other models present no advantages.
In figure 2 we present a boxplot of the weighted accuracy of the learning techniques.

	Accuracy Mean	Accuracy SD	W. Accuracy Mean	W. Accuracy SD
Tree	62.94%	7.23%	62.29%	6.13%
Random Forest	76.76%	4.25%	77.37%	3.38%
Linear SVM	84.71%	6.14%	86.08%	6.29%
Polynomial SVM	73.24%	5.49%	74.94%	6.14%
Gaussian SVM	77.65%	4.40%	78.66%	3.98%
kNN	62.35%	6.94%	62.62%	7.60%

Figure 1: Results for selected learning techniques

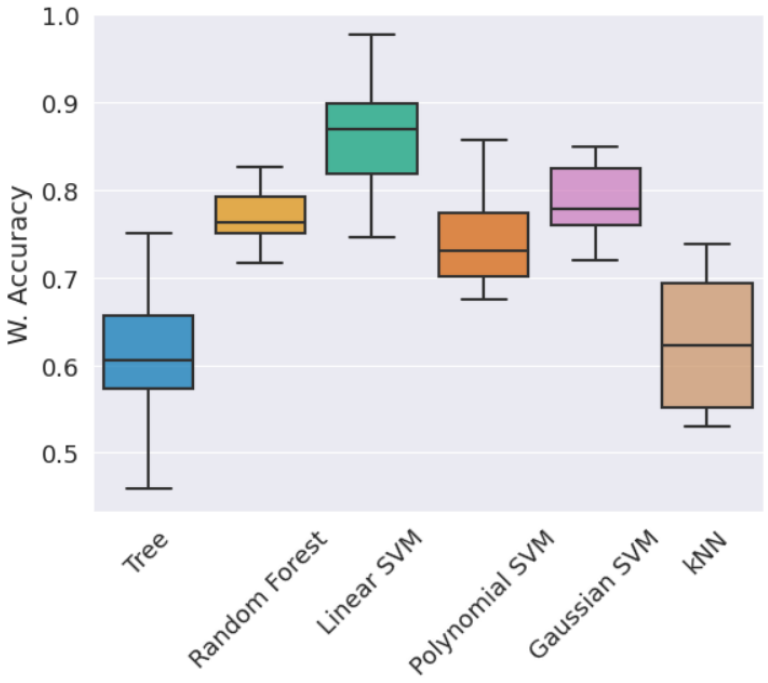

Figure 2: Boxplot of Weighted Accuracies

In the table below, we report the time taken to perform hyperparameter tuning plus the time to fit the model to the entire dataset, in seconds. In terms of efficiency, the linear kernel SVM is better than the Random Forest.

Model	Time
Tree	1.66s
Random Forest	16.80s
Linear SVM	8.86s
Polynomial SVM	17.70s
Gaussian SVM	8.41s
kNN	5.15s

5 Conclusions

The considered effectiveness indexes suggest to use an SVM with a linear kernel to tackle the problem, using the parameters indicated in paragraph 4.2.
It should be noted that the used dataset is small with some classes comparing in as few as 8 observations, and might not be representative of the real system. If a new dataset can be gathered, it is suggested to perform again hyperparameter tuning for the SVM, and do a comparison with the hyperparamter-free Random Forest.

References

[1] Eric Medvet. Introduction to machine learning and evolutionary robotics. https://medvet.inginf.units.it/teaching/2223-intro-ml-er/, 2022-2023.

[2] scikit learn. Svc. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.

[3] Pedro F. B. Silva, André R. S. Marçal, and Rubim Almeida da Silva. leaf dataset. https://archive.ics.uci.edu/ml/machine-learning-databases/00288/, 2014.

[4] Pedro F. B. Silva, André R. S. Marçal, and Rubim M. Almeida da Silva. Evaluation of features for leaf discrimination. In Mohamed Kamel and Aurélio J. C. Campilho, editors, *Image Analysis and Recognition - 10th International Conference, ICIAR 2013, Póvoa do Varzim, Portugal, June 26-28, 2013. Proceedings*, volume 7950 of *Lecture Notes in Computer Science*, pages 197–204. Springer, 2013.