Fraunhofer
PORTUGAL

ML4MALARIA

STATISTICAL COMPARISON OF DIFFERENT MACHINE LEARNING APPROACHES FOR MALARIA PARASITE DETECTION IN MICROSCOPIC IMAGES

Motivation

Malaria is a severe public health problem across the world, particularly in developing countries (≈80% of the cases occur in Africa), putting at special risk the most unprotected groups of society: children and pregnant women. Since it can be caused by 4 different species of parasites, each having different stages of evolution, approaching the right diagnosis without access to costly equipment is complex. Ergo, research has focused on speeding up and lowering the costs of its diagnosis, by resorting to automatic machine classification of microscopic images. Still, most approaches rely on a simplistic, single-model classifiers, with a constant absence of a systematic statistical comparison in the literature that supports a particular technique or feature.

Goals

Having as basis the *MalariaScope* dataset and results, this main objective for this project is to **derive a systematic method that can**:

- Explore a wide space of *Feature Selection* methods, *Machine Learning Classifiers*, and tuning the *hyperparameters*, to provide a classifier able to identify the presence of Malaria in microscopical images;
- Provide statistical relevance of the performance of each obtained classifier, when compared to others, by relying on robust metrics.

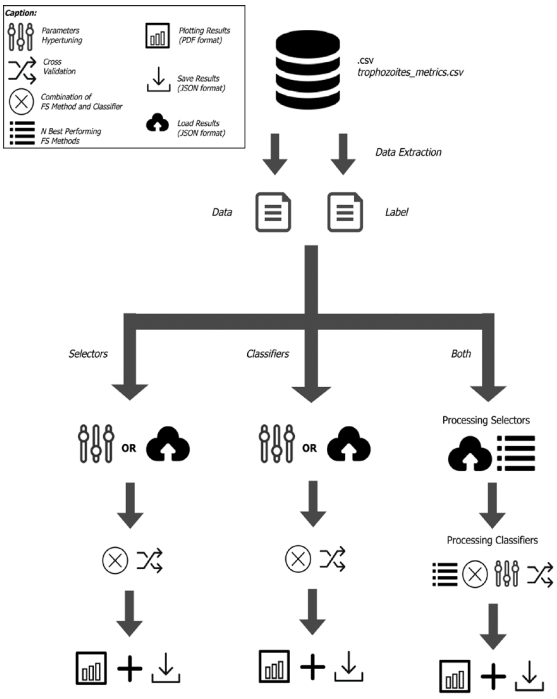

Fig1. Developed Machine Learning Framework Design.

Framework

The framework was developed in *Python*. To use it, one must one the command line terminal. The main usage of the tool can be summarized by invoking it with the help `-help` argument. This tool presents three main

Contact

Rua Alfredo Allen, 455
4200-135 Porto, Portugal

+351 220 430 300
info@fraunhofer.pt
www.fraunhofer.pt

Framework Main Features

- *Hyperparameter* Optimization for the Feature Selections and Classification Algortihms
- Cross Validation
- Statistical Hypothesis Test
- Charts creation for Information Visualization

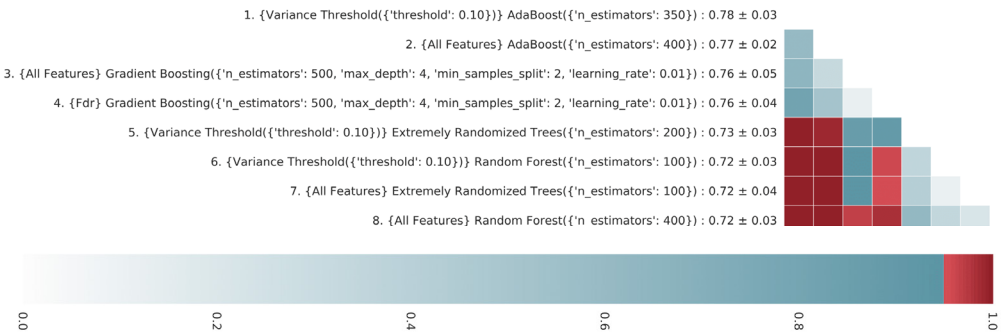

Fig2. Statistical Hypothesis *T-Test* Heatmap Matrix of the Final 8 Results.

paths, namely: (i) **Feature Selection**; (ii) **Classification**; and (iii) **Analysis**.

Initially, the user supplies the wanted dataset, and then he can:

(i) `-sel`, `--selectors`: Analyze which are the best feature selection methods, by *hypertuning* its parameters and using few classifiers with the default settings;
(ii) `-clf`, `--classifier`: Analyze which are the best classifiers applied to the dataset, by *hypertuning* its parameters and using some default feature selection methods. These methods were selected after testing, being picked the ones that usually provide the best results;
(iii) `-all`: Analyze which are the best combos, i.e., the combination of feature selection methods and classifiers. It should be noted that it is possible to define the number of the selectors that will feed the classifiers (which is 10, by default), with the optional argument `-n`, `--nselectors`.

After choosing a path option, but before starting the processes, the user has also several optional arguments:

(iv) `-s`, `--save` serializes the intermediate results to a *.json* file;
(v) `-l`, `--load` loads previously serialized results from a *.json* file;
(vi) `--seed` defines the seed of the random number generator for the data shuffle process;
(vii) `--viz` chooses the visualization output;
(viii) `--mtr` selects the metric used on the cross validation process.

Results

The results analysis process starts by studying the best performance metric (according the dataset proportions), as well by computing a statistical hypothesis test to prove that the selected data model is, in fact, the best. For this domain, it was conclude that both feature selection and different classifiers – with corresponding *hyperparameters* tuning – can lead to better results according to the evaluated F_1 *score*. In particular, selecting the most relevant features using **Variance Threshold with *threshold = 0.10*** (subset of 200 features), and subsequently training an **Ada Boost ensemble classifier with *n_estimators = 350***, with a F_1 *score* of 78%, leads to a statistically significant better result than previously obtained (F_1 *score* = 75%), as well as when compared to other combinations of selectors, classifiers, and *hyperparameters*.