MAC0460/MAC5832 – EP4

DCC / IME-USP — Primeiro semestre de 2021

Objectives: The aim of this EP is to

- Practice training of linear, neural networks, and SVMs classifiers using the `scikit-learn` library (`https://scikit-learn.org/`)
- Practice model evaluation, comparison and selection
- Produce a summary report on the performed experiments and main findings

Description

In this EP we will use the MNIST dataset to accomplish the objectives listed above. We will work with three models (algorithms): logistic regression, neural networks, and SVM. You will use the implementation available in the `scikit-learn` library (`https://scikit-learn.org/`).

This time, there is no notebook to be filled. You will need to write all the code and comment the code whenever you judge comments help understanding.

A good experimental practice is to write codes that allow reproduction of the results. Therefore, whenever your code depends on random processing, be sure to use a fixed seed value.

You should submit a notebook with the implementation of the steps detailed below, with the outputs of your code cells. At the end of the notebook, you must add a section with an overview of the implemented experiments, results and relevant comments (it is likely that some comments will be already spread along the notebook; there is no problem if there is repetition of part of the comments). Feel free (actualy, we encourage you) to add prints, plots, or any other resources that enrich the analysis of the results.

1 Dataset preparation

A notebook for reading the MNIST dataset using the `tensorflow.keras` library is provided (ep04.ipynb). The notebook also includes a preprocessing for reducing the image size as well as normalizing the intensities to the interval $[0, 1]$.

You are free to rewrite the code and fetch the MNIST dataset from other sources if you wish, but you must employ the same preprocessing done in the provided notebook.

MNIST consists of 60000 training and 10000 testing images. You will not touch the testing set until the very end.

You must partition the original training set into 70% training (D_{train}) and 30% validation (D_{val}) sets, in a stratified way. Make sure to include code to verify that the generated partition is indeed stratified.

2 Training, evaluating and selecting models

You will use D_{train} to train and select three models: a logistic regression model, a neural network model, and a SVM model.

The training/evaluation/selection process must use only data in D_{train}. Each of these three algorithms have hyperparameters. You may fix or test different values for the hyperparameters. Whether further partition of D_{train} is needed or not, this is left to your judgement.

Overall, it is expected that you will implement a model selection method. For that, you can use implementations available in the `scikit-learn` library. The model selection can be based on performance metrics of your choice.

You must explain the method you are using to select the models, including comments on which hyperparameters you are evaluating (if you used solutions avaliable elsewhere as references, please provide full information about them).

Clearly indicate and justify the selected three models.

Remark: Regardless of the performance metrics you will consider, the value computed by the `score` method MUST be printed for all classifiers (not only in this step but also in the next steps).

3 Choosing a final model

In this step, the goal is to evaluate – on the validation set (D_{val}) – the three models selected in the previous step.

Compare the performance computed with respect to (D_{val}) with the ones obtained in the previous step. Comment the comparison results. In particular, it is expected that the linear model will work as a baseline. Comment the performance of the other models when compared to the linear model.

Clearly indicate and justify the chosen final model.

4 Error estimation

Now that you have chosen the final model, you can compute an estimate of its expected performance (E_{out} and other metrics/analysis). Note that this is the only part of your code where you will use the test set.

The model you will evaluate here on the test set is the one that presented the best validation error with respect to D_{val}. Therefore it might be interesting to compare how similar or how different are the performance metrics computed on D_{test} and D_{val}.

You may also retrain the selected model on $D_{train} \cup D_{val}$ and verify if there is any significant difference.

EP evaluation criteria

The EP will be evaluated based on code organization (clarity of presentation), fulfillment (implementation of the steps listed above), correctness (concepts applied correctly, code implemented correctly), value (clarity, quality and completeness of evaluation and analysis).