UCLouvain

École polytechnique de Louvain

LINFO2262 : Machine Learning

"A Machine Learning Competition"

Professor : P. Dupont

Robin Paquet

May 2022

1 Introduction

This is a short report containing the choices that have been made for the realization of the project. This includes the tools and softwares that have been used to perform the classification.

2 Algorithms/software

I used several libraries to complete this project :

— Scikit-learn contains Python implementations of a large number of standard machine learning algorithms.

— NumPy is the most commonly used Python library for scientific computing.

— Pandas is used to represent data under the form of data frames, a structure which allows you to easily manipulate your data.

3 Data preprocessing

In terms of data preprocessing, we notice when we go through the given training set and the test set, that it is necessary to go through a few preprocessing steps to exploit the data optimally in the model.

The first operation was to find each of the *NaN* values and replace them with the median which is the midpoint of the set so as not to distort the results. Another solution would have been to replace the values with zeros.

Then next operation was to replace the cells containing strings with numbers since many machine learning algorithms cannot operate on label data directly has it is the case for *scikit-learn*. They require all input variables and output variables to be numeric. For this the *OrdinalEncoder* was useful, it allows to encode categorical features in the form of an integer array.

4 Classification model

After doing some research to find the right estimator, I was interested in the *Stochastic Gradient Descent*. Indeed, this model seems to be appropriate for the format and the high amount of data in the dataset.

Stochastic Gradient Descent (SGD) is a simple yet efficient optimization algorithm used to find the values of parameters/coefficients of functions that minimize a cost function. In other words, it is used for discriminative learning of linear classifiers under convex loss functions such as SVM and Logistic regression. Also *Stochastic Gradient Descent* is very efficient and it is easy to implement as there are lots of opportunities for code tuning.

5 Hyperparameter tuning

In order to obtain the best prediction, it was necessary to modify the hyperparameters of the classification model. To make the choice of these hyperparameter, I used the parameter optimization tool offered by *scikit-learn* which is called *RandomizedSearchCV*. It is a tool that works through random search on parameters. Unlike gridsearchCV which tests all parameter values, making it much slower with its search on all combinations. To overcome this, I therefore carried out the random search several times, keeping the parameters that gave the best score.

6 Model evaluation

As we have seen, every estimator exposes a score method that can judge the quality of the prediction on new data. To get a better measure of prediction accuracy, we can successively split the data in folds that we use for training and testing. So to estimate the skill of the machine learning model, I used the *StratifiedKfold* cross-validation method.

Indeed, when the target variable is of binary or multiclass type, it is this technique that is used. *StratifiedKFold* is a variation of k-fold which returns stratified folds, each set contains approximately the same percentage of samples of each target class as the complete set.

Since the dataset needs not be perfectly balanced in terms of class priors, the chosen test performance metric of a classifier is defined as the balanced classification rate (BCR). For this, I used the *balanced_ accuracy_ score* proposed by *scikit-learn* in combination with the data split to computes the classification rate for each class