EZStacking 0.5

October 15, 2021

Abstract

EZStacking is a Jupyter notebook generator that produces a graphical exploratory data analysis and a modelling of the input data. With the GUI, the machine learning is made easy.

1 Prepare your environment

This small project is developed under Ubuntu Linux , based on Anaconda Python Environment. The Python version is 3.8.12. If you use Anaconda, it is easy to install the Jupyter notebook, unfortunately you have to install the following packages too :

1. for computations:
    - pandas 1.3.3
    - scikit-learn 0.24.1
    - keras 2.4.3
    - xgboost 1.3.3
    - polylearn 0.1
    - scipy 1.6.0
2. for graphics:
    - yellowbrick (developement version)
    - matplotlib 3.4.2
    - seaborn 0.11.2
    - graphviz 2.40.1 & python-graphviz: 0.16
    - nbformat 5.1.2
    - ipywidgets 7.6.3
    - ipyfilechooser 0.6.0

2 Install EZStacking

Once the archive is downloaded, you just have to unzip it in the folder of your choice.

3 How to use EZStacking

Open the folder in Jupyter:

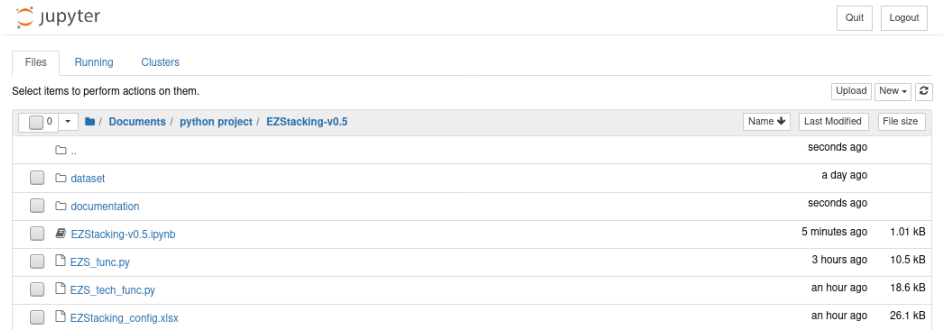

Figure 1: EZStacking folder

Then open the notebook EZStacking-v0.5.ipynb:

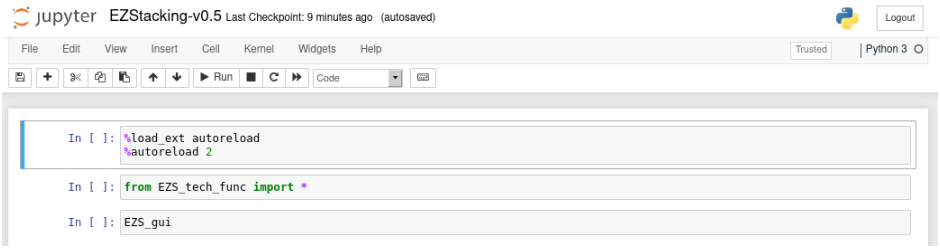

Figure 2: EZStacking notebook before running

3.1 EZStacking GUI

Now it is time to run all notebook cells:

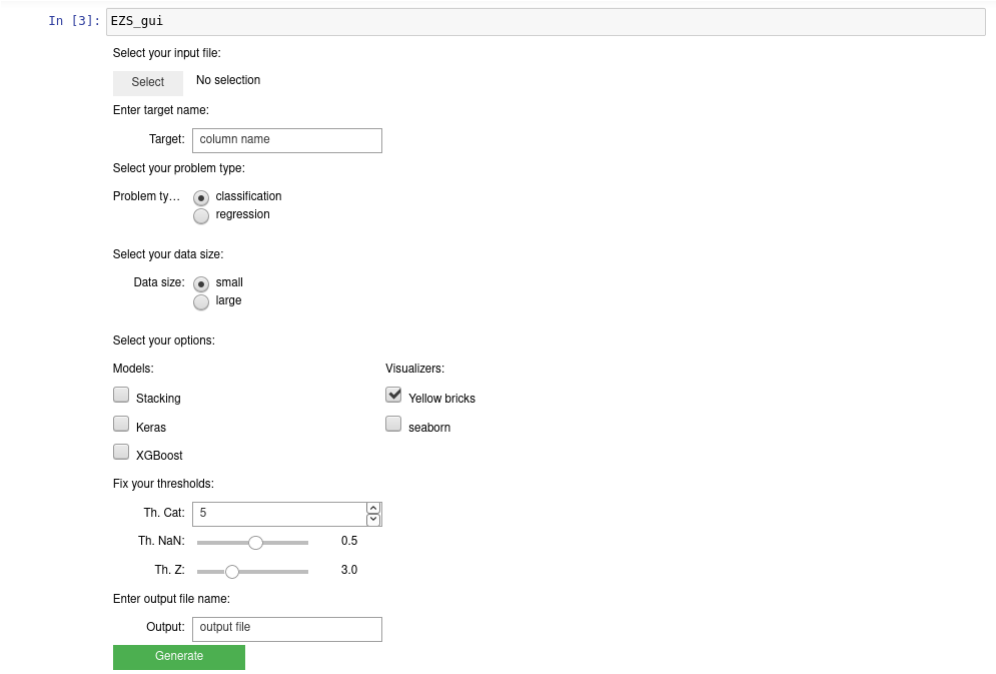

Figure 3: EZStacking GUI

3.2 Settings

First click on "Select" and select a file in the dataset folder (e.g. iris.csv or concrete.csv):

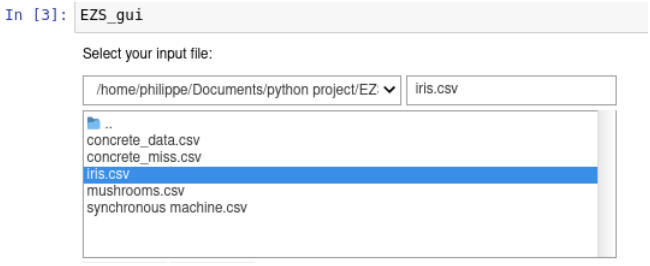

Figure 4: File selection

Then input the target name (e.g. variety or Strength):

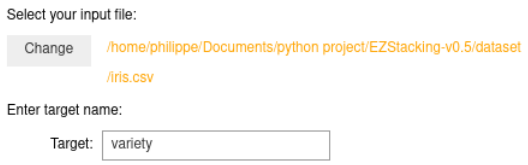

Figure 5: Target input

Here the problem type[1] is "classification" (there are three types of iris), the data size[2] is small:

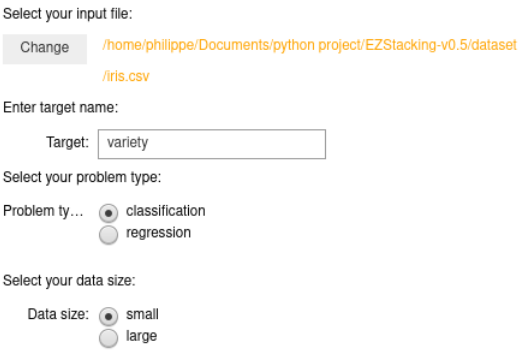

Figure 6: Problem type & Data size

Note : a dataset should be considered as "small" if $n_{row} < 3000$.

[1] if you use the dataset concrete.csv, the problem type is "regression" (the Strength is not categorical)

[2] if you use the dataset concrete.csv, the data size should be "large"

Let's fix the processing options (here full options):

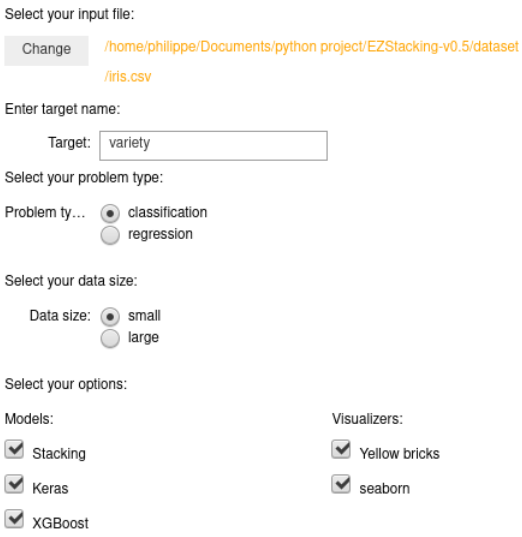

Figure 7: Processing options

Option	Description	
Stacking	When activated, the final model will use the "Stacked generalization"	
Keras	With Stacking	A built-in Keras neural network is added to the stack
	Without	A built-in Keras neural network is used as unique model
XGBoost	With Stacking	A built-in XGBoost model is added in the stack
	Without	A built-in XGBoost model is used as unique model
Yellow Bricks	Using Yellow Bricks, you will add many graphics to you notebook	
Seaborn	Seaborn should not be used with too large dataset...	

Table 1: Processing options in detail

Notes :

- *if the Stacking option is unchecked, you should not check Keras and XGBoost options, only Keras will appear in the final notebook.*
- *if no option is checked (or just "Stacking"), it will generate a notebook based on Scikit-Learn.*

Last step, the thresholds :

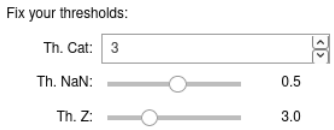

Figure 8: Thresholds

Notes :

- *threshold_cat: threshold for categorical data, if the number of different values in a column is less than this number, the column will be considered as a categorical column.*
- *threshold_NaN: hreshold for* `NaN`*, if the proportion of* `NaN` *is greater than this number the column will be dropped*
- *threshold_Z: threshold for outliers, if the Z_score is greater than this number, the row will be dropped*

3.3 Generation and submission

Now fill the output file name (e.g. iris_full) and click on the button "Generate". If no error message, you should find a new file named iris_full,ipynb in the Jupyter main page:

Figure 9: Jupyter main page

Now just select and run it... The results are not displayed here, but they can be found in the zip archive.

4 Models used in EZStacking

According to papers about stacked generalization, the diversity of models is crucial, so EZStacking uses the following types of models:

- classical regressions (linear or logistic)
- stochastic gradient descents (that extends regression for larger datasets)
- k-nearest neighbours
- support vector machines
- Gaussian processes
- neural networks (Keras or Scikit-Learn)
- factorization machines and polynomial networks
- decision trees
- ensemble based on boosting.