LAB 2: CLASSIFICATION IN DATA STREAMS

10 January 2018

The goal of this lab is to build a classifier capable of learning and making predictions in a data-stream. The code in `run_experiments.py` loads the Electricity dataset and uses it to evaluate three data-stream classifiers:

- **kNN**
- **Hoeffding Tree**
- **Batch-Incremental Ensemble Classifier (BIE)**

except the last of these is not yet (properly) implemented. That is the task of this lab. You should implement the `predict` and `partial_fit` functions in the outline of the classifier in `my_classifier.py`. Implement a tumbling window of size 100, creating and maintaining up to a maximum of 100 models. Build a `DecisionTreeClassifier` on each of the batches/windows.

If you run the script `run_plot.py` you will obtain a result like that in Figure 1. After you implement the BIE classifier, you should run this script again, and submit (within a zip file with your name, e.g., `Firstname_LASTNAME.zip`):

1. The resulting figure (`result_elec.pdf`) along with
2. Your code `my_classifier.py`,
3. A text file with the output of the final lines from `run_classifier.py` (*after* implementation) which include the `Evaluation time` and `Global accuracy` of the methods.

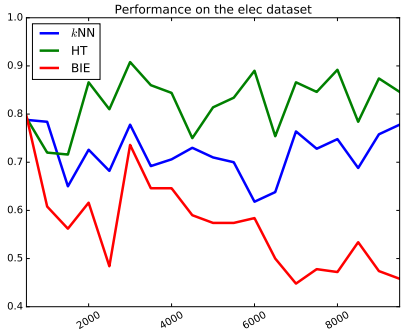

Figure 1: Evaluation over time on the Electricity dataset.

Software Requirements: The lab requires SCIKITMULTIFLOW[1], a data-stream learning framework in early development, based on SCIKITLEARN and its dependencies (e.g., NUMPY). Also PANDAS and MATPLOTLIB. The lab can be run with either the Python 3.X (recommended) or 2.X interpreter.

[1] `https://github.com/scikit-multiflow`