

GE461: Introduction to Data Science

Assignment for Data Stream Mining

April 28, 2025

Due date: May 14, 2025 Thursday, 23:59

Late Policy: One day late (any time on May 15) grading will be out of 90

Two days late (any time on May 16) grading will be out of 80

Three or more days late: No submission will be accepted.

Notes: In this assignment, you will build an evolving data stream classification model. You will use scikit-multiflow, an open-source Python library that provides tools for data stream mining and classification. You can find more information about this library on the official webpage (<https://scikit-multiflow.github.io/>). Your task is to train data stream classification methods on the given data streams. You will use test-then-train –i.e., prequential evaluation method to compare performance of the classification models.

For installing scikitmultiflow, create a new environment and install the requirements listed in the following link.

Requirements link: [scikit-multiflow/requirements.txt at master · scikit-multiflow/scikit-multiflow · GitHub](https://github.com/scikit-multiflow/scikit-multiflow/blob/master/requirements.txt)

Teaching Assistant: Sepehr Bakhshi, sepehr.bakhshi@bilkent.edu.tr

TA Office Hour Administration: Please send your question to Sepehr. He can arrange a meeting upon request.

A. What to Submit

Your submission has two components.

1. **Code.** It must contain proper comments. You must also include your name at the top as a signature that confirms that you are the programmer. Please remember that MOSS is in our plans for plagiarism check.
2. **Report.** Your report must be in pdf form and must cover all "Work to be Done" sections of the assignment (except part C.1 and C.2). Explicitly write the steps that you have taken in each section. Please also write a brief introduction that includes the overall structure of the report (You can use introduction part of the scientific papers as an example to write a similar one for this homework). For each section of your work explain the purpose and what has been done and achieved in that section. Provide a comparison of results that contains tables and plots as appropriate. Make sure that you follow the principles of scientific writing. Use simple past or simple present tense in your report. If you plan to propose future work then in that case you may use future tense.

Your report must have proper title like a scientific paper, reflecting its true content. It must have your name and address etc. If you like, for experience and fun, you may use the ACM conference paper format¹. You must use latex or Microsoft Word or their equivalent.

Optional: As an optional part, at the beginning of your report, you may have a related works section that covers data stream mining briefly with proper references.

Optional: Another optional part is comparison of the effectiveness of the methods using statistical tests. The design and administration of these tests should be decided by you by looking at the available papers in literature.

See Justin Zobel's book *Writing for Computer Science* for further hints on the style of CS related scientific paper writing.

B. Submitting Your Work

You will submit your work by uploading it to Moodle in a zipped file. Its name must be streamMiningYourFirstNameYourLastName. For a student with the name "Ali Can Ok" it is streamMiningAliOk.

C. Work to be Done

¹ https://www.acm.org/binaries/content/assets/publications/taps/acm_layout_submission_template.pdf

1. Concepts:

- *Data Stream*: refers to an environment where data arrives continuously over time. This means we cannot assume that we have access to all the data at the beginning. Instead, we need to update our model incrementally as new data arrives. This is in contrast to traditional batch learning, where we can access all the data simultaneously and train the model on the entire dataset.
- *Concept drift*: refers to a change in the underlying distribution of the data. For example, in the case of a spam email detection model, the characteristics of spam emails may change over time, which means that the system needs to adapt and update itself to identify the new characteristics of spam emails correctly. Data stream classification models need to adopt a concept drift detection and handling method to address concept drift. We refer to a data stream with concept drift as evolving data stream.
- *Prequential Evaluation*: is a commonly used evaluation approach for data stream classification tasks. In the prequential evaluation method, the model is tested on each incoming instance before it is used to update the model. It is also known as interleaved test-then-train evaluation method.

2. Requirements:

You will need the following libraries in Python to complete this challenge.

- numpy: a Python library for numerical computing
- scikit-learn: a library for machine learning in Python
- scikit-multiflow: a library for data stream mining and classification

3. Datasets as Data streams

You will use two synthetic and two real datasets as data streams to compare performance of the classification models. As synthetic data streams, use AGRAWALGenerator and SEAGenerator classes from scikit-multiflow to generate 100,000 data instances for each. For future access, write the generated data instances into files named AGRAWALGenerator and SEADataset.

Each synthetic dataset must have two abrupt drift points (drift width = 1) at positions 35k and 60k. To apply concept drift on these datasets, please read the documentation of [SEAGenerator documentation](#).

As real datasets, you will experiment with the Spam and Electricity datasets. You can obtain these datasets from <https://github.com/ogozuacik/conceptdrift-datasets-scikit-multiflow>.

4. Classification Task with Concept Drift Handling

4.1. Implement an instance of the following classification models. You can use scikit-multiflow to this aim.

- Adaptive Random Forest (ARF) [4]
- Streaming Agnostic Model with k-Nearest Neighbors (SAM-kNN) [3]

4.2. Each base learner in your ensemble must be associated with a different drift detector. Use a combination of detectors such as DDM, EDDM, and ADWIN, assigning them either randomly or following a round-robin strategy.

Implement adaptive weighting for combining the predictions of the base learners. Specifically, assign weights to the learners based on their recent prediction accuracy measured over a sliding window of fixed size (e.g., 100 recent instances). The weighted majority vote should be used for the final prediction.

When a concept drift is detected by a learner's associated detector, replace that specific learner with a new HoeffdingTreeClassifier, and reset its associated drift detector and sliding window statistics.

Hint: You can use drift detection mechanisms included in scikit-multiflow such as DDM, EDDM, and ADWIN.

4.3. Extend your custom ensemble to implement passive drift detection. In passive detection, the ensemble should infer the presence of concept drift internally without relying on an external drift detector.

Design a simple passive detection strategy. For example, if the sliding window accuracy of a base learner drops below a predefined threshold (e.g., 70%), treat this as a signal of concept drift for that learner.

When passive drift is detected, replace the affected learner with a new instance of HoeffdingTreeClassifier, similar to the active version.

Note: The ensemble approaches in Sections 4.2 and 4.3 must be implemented from scratch. You are only allowed to import the HoeffdingTreeClassifier, your preferred drift detectors, and basic libraries (e.g., numpy, pandas). Do not use any ensemble constructors from scikit-multiflow. Submissions that violate this rule will receive zero points for these sections.

5. Results and Discussion

Construct an instance of the classification models. For each dataset, use Interleaved Test-Then-Train approach to train and evaluate performance of these classifiers. Use prediction accuracy as evaluation metric. Report the following results for the classification models on each dataset:

- Overall accuracy: Overall prediction accuracy of the models.
- Prequential accuracy plot: Prequential accuracy is defined as the prediction accuracy of a model over the w most recent data instances. Use 20 sliding windows of size (dataset size/20) to calculate prequential accuracy values. Plot the obtained accuracy values over time for each dataset.

5.1. How does your ensemble model perform compared to the state-of-the-art approaches in 4.1? What could be possible improvements for a more robust ensemble. Discuss your findings on the accuracy plots. What is inferred from the drops in the prequential accuracy plot?

5.2. Try different window sizes in your ensemble. How does it affect the prediction accuracy?

5.3. Compare the Active and Passive versions of your custom ensemble. Indicate clearly which version achieved higher accuracy. Discuss the observed results: Why do you think one ensemble outperformed the other? Consider aspects such as reaction speed to drift, false alarms, sensitivity to noise, and model stability when reasoning about the results.

5.4. Please also include a paragraph that summarizes your findings in this assignment. What did you learn from this assignment?

In your comparisons use plots and tables when appropriate. Number all plots and tables and provide proper subtitles for them. Make sure that you refer to each of them in the text of your report. Help your reader by providing a simple and easy to follow presentation.