3945 – Advanced Machine Learning, Semester A 2025
Home Assignment 2: Boosting

Lecturer: Dr. Leon Anavy

In this assignment, you will implement AdaBoost. You will demonstrate the models and design experiments to test their performance.

You should submit a single ipython notebook (.ipynb) file that includes all the code and outputs.

You are encouraged to use methods and techniques covered in class but are also welcome to use any other relevant method you find suitable for the task if you justify your decision.

Make sure to use proper ML and DS methodologies when needed (Data preprocessing, Data splits, etc.).

Submission Guidelines

- Submit a single Jupyter notebook (.ipynb) titled **HW2_ID1_ID2.ipynb** that includes:
    - Appropriately formatted cells (markdown) describing the design, implementation, and showing any calculations.
    - All code and outputs.
    - Explanations for each step.
    - Observations and conclusions for each section.
- Ensure your notebook runs from start to finish without errors.

You may use external libraries as needed. Specify all dependencies in the notebook.

Grading

- The work will be assessed based on methodology, execution, and presentation.
- All figures and plots should be correctly labeled.
- Grading will be based on correctness, elegance of solution, and style (comments, naming conventions, etc.).

Part 1: Design the Algorithm

- Describe the design you will use to implement Adaboost. Clearly describe the theoretical and practical considerations for the algorithm, any limitations, and discuss use-cases, as well as hyperparameters, and any optimization strategies if relevant.
- Ensure that your design captures the initialization, weight updates, and iterative boosting steps.

Part 2: Implement the Algorithm

- Based on your design, implement the Adaboost algorithm.
- Write **efficient**, **vectorized** code whenever possible.
- You must explain and justify the design.
- Handle hyperparameters correctly.

- Develop your implementation of AdaBoost from scratch, except for using library functions for mathematical calculations or weighted decision trees.

Part 3: Demonstrate the Algorithm

- Test your AdaBoost implementation on a simple, simulated dataset (a "unit test") to validate correctness and debug any issues.
- Justify the simulated dataset's characteristics.
- Justify the parameters you set for the algorithm.
- Evaluate your implementation by comparing its outputs to those of an existing AdaBoost implementation.

Part 4: Experimental Design and Analysis

- Generate **at least** two experimental datasets with binary labels, designed to demonstrate specific properties of AdaBoost (e.g., handling noise or overfitting).
- Split each dataset into training and test sets.
- Visualize the data (dimensionality reduction is allowed if necessary).
- Perform experiments to evaluate your AdaBoost implementation using the generated datasets.
- In addition, test a library implementation of AdaBoost and **at least** two additional models, one of which must be another boosting algorithm.
- Run short experiments on the algorithms and their hyperparameters, and use proper performance metrics to evaluate them, justifying your choice of metrics.
- Present your results and evaluate the impact of the dataset properties on the performance of your models.

Experimentation Guidelines

- Clearly describe the experimental setup, including:
  - Dataset characteristics.
  - Data preprocessing steps.
  - Parameter selection.
  - Justification for key design and implementation choices.
- Use appropriate plots to demonstrate results.
- Discuss the observations and insights derived from the results.

Important Note Regarding Generative AI: While you may use Generative AI models to assist you, ensure all submitted work is original and that you fully understand it. You may be asked to explain your work in an in-person review.

Pair Work: Submit in pairs, with one submission per pair.