Final Project

For your final projects, you will **work in a group** to implement computational/ machine learning algrithms **from scratch** using what we have talked about in class. You can either implement an algorithm we have discussed in the lecture (but without using the exactly same code provided) or any method you find interesting/ relevant to your research. After the implementation, you are also required to use Shiny to launch a dashboard to present your work.

Collaboration Guidelines

1. You may work in a **group of two or three students**.
2. If you work in a group of **two**, you are expected to implement **at least two algorithms**.
   If you work in a group of **three**, you are expected to implement **at least three algorithms**.
3. The algorithms you choose should be **conceptually related** rather than completely independent.
   For example, avoid mixing linear regression, logistic regression, and decision trees within the same project.
   Instead, focus on a consistent theme—such as
   - variations or enhancements of boosting methods,
   - efficient matrix compression techniques, or
   - algorithm acceleration and optimization strategies.
4. You are encouraged to explore algorithms **beyond those covered in lectures** if they fit your chosen theme.

5. **Avoid implementing overly simple or trivial algorithms** that do not involve meaningful computation or algorithmic design.

   Examples of projects that would be **too straightforward** include:

   - Directly coding simple linear regression or ordinary least squares.
   - Implementing basic mean/variance computations or correlation analysis.
   - Writing a standard k-means function without any modification, optimization, or extension.

   Instead, aim for algorithms that demonstrate your understanding of **optimization, iteration, convergence analysis, or computational efficiency**.

6. Each team must submit the following deliverables:
   - A **written report** (documenting objectives, methods, and results)
   - **Presentation slides**
   - A **Shiny dashboard** showcasing your results
   - Supporting **code and datasets**

Code Requirements

1. You may choose your preferred programming language; **R** or **Python** are recommended, while **Rcpp** is optional.
2. The algorithm you implement may be either **covered or not covered** in the lectures. However, you **must not reproduce** the exact same code provided in class materials.
3. The **core logic** of your algorithm must be **implemented from scratch** (for example, do not use built-in modules such as sklearn.svm when implementing an SVM).

You may, however, use existing libraries for **data preprocessing, standardization, and visualization**.

4. After implementation, **test and validate** your algorithm using either **synthetic** or **real-world datasets** to ensure correctness and robustness.
5. Your project will be evaluated based on the **creativity**, **originality**, **accuracy**, and **thoroughness** of your implementation and analysis.

Shiny Dashboard requirement:

Use your selected data to create a Shiny app or dashboard that allows users to interact, explore, and better understand the data. You may want to work with the Navigation Bar Page layout or Shiny Dashboard. Both options will neatly structure your app's user interface; however, these are not the only options.

Required app features:

1. An information section - what is the data, where did it come from, what is the purpose. Make it clear to users what data they are exploring.
2. It should have at least three different input/control widgets.
3. It should have at least three different output displays.
4. It should be well organized and aesthetically pleasing.
5. (optional) Feel free to incorporate at least one derivative shiny package such as shinythemes or shinyalert. There are many more, these are just two examples.

Check out Shiny's gallery for inspiration.

You will write up your analysis in a written report, and make an oral presentation. The presentation will be at most **15 minutes** in total.

Expectations for Your Presentation and Report

1. **Introduction:**

   Begin with a clear introduction to the algorithm you have implemented.

   - If the algorithm was discussed in lecture, keep this section concise and focus on your own contribution or modification.

2. **Motivation:**

   Explain the motivation behind your project — why you selected this algorithm and what question or challenge you aim to address.

3. **Theory and Implementation:**

   Describe the **theoretical foundation** of the algorithm and its **numerical implementation** in detail.

   - Clearly outline how each step of the computation is performed, emphasizing your own design and coding process.

4. **Evaluation and Results:**

   Test and evaluate your algorithm using **synthetic or real datasets**, and present your results clearly.

   - Use figures, tables, or performance metrics to illustrate and support your findings.

5. **Discussion:**

   Provide a critical discussion of your algorithm and implementation. For example, you may:

- Comment on the **efficiency** and **stability** of your algorithm.
- Compare your approach with **existing implementations** in the literature.
- Suggest **potential improvements** or future extensions of your work.

6. **Speculation and Language Use:**

   When interpreting results, use cautious and evidence-based language.

   - Prefer expressions such as *"It seems," "It appears,"* or *"The residuals appear approximately normal"* rather than absolute statements like *"The residuals are normal."*

7. **Written Report:**

   Your written report should include the following sections: **Introduction**, **Algorithm and Implementation**, **Results**, and **Discussion**.

   - The main text should be **no more than five pages**, followed by any number of figures, tables, or appendices.

8. **Oral Presentation:**

   In your presentation:

   - Start with an overview of the algorithm you implemented.
   - Explain your computational design and implementation steps.
   - Present and evaluate your results — **visualizations are strongly encouraged**.
   - Conclude with key findings and reflections.
   - Showcase your Shiny app and describe all the capabilities.

9. **Submission:**

   Submit your **written report**, **code scripts**, and **presentation slides** by **December 8**.

- You may make minor revisions to your slides before the oral presentation; please ensure the updated version is uploaded to **Canvas** before your presentation session.