Instruction for Assignment 2

EE214
(2025 Spring)

May 6, 2025

1 Overview

In this assignment, you will practice full end-to-end unsupervised learning workflows using classical clustering algorithms which are covered in our lectures. The assignment is structured in three parts as below. You are allowed and encouraged to use AI tools (e.g., ChatGPT) to help with coding and understanding, but you must verify all answers and document your usage of AI. The goal is to ensure you learn the concepts and not rely on AI to do everything. The assignment includes:

- **Part 1: Fill-in-the-Blanks**: Complete a provided Jupyter Notebook with key tasks including preprocessing, clustering algorithms and visualization tasks.
- **Part 2: Coding Task**: Tackle a more open-ended clustering problem that requires deeper understanding of the workflows.
- **Part 3: Report**: Write a short report explaining your approach, results, and how you used AI, including what was helpful and what limitations you encountered.

Each part builds on the previous one in difficulty. By the end, you will have experience with a full ML pipeline and reflection on the problem-solving process.

Using AI tools: You may ask ChatGPT or other AI for help (e.g., debugging code, explaining concepts, suggesting approaches). However, the final work must be your own:

- Do not blindly copy AI outputs, always test and understand any code or answers given.
- Document in your report exactly how you used AI (which tasks, what it helped with, where it failed).
- Keep in mind that AI can be wrong or incomplete. Part of your grade is evaluating AI's answers critically.

This approach is designed to enhance learning: reflecting on your thought process and AI's role can deepen your understanding. By explaining challenges and how you overcame them, you engage in higher-level thinking that AI cannot do for you.

What to Include

- **Approach and Solution Outline**: Narrate, in your own words, the steps you took for Parts 2.1 and 2.2, including any debugging or parameter tuning.
- **Results and Analysis**: Summarise key findings—chosen cluster numbers, best algorithms, notable misclusterings—and what you learned about clustering metrics.
- **Use of AI**: A mandatory disclosure specifying the tools used, tasks assisted, integration of suggestions, and any incorrect AI output you corrected.

2 Report

You will write a report describing your work and your use of AI. This report is a crucial component of the assignment for reflecting on what you learned and ensuring a fair evaluation and transparency in using AI for academic work.

2.1 What to Include in the Report:

Approach and Solution Outline: Describe in *your own words* how you tackled Part 2 (and Part 1, if relevant). Give a clear narrative of your workflow—e.g., how you pre-processed the data, selected candidate algorithms, decided which internal metrics (Elbow, Silhouette, dendrogram, Davies–Bouldin, etc.) to trust, and how you selected hyper-parameters (such as k or DBSCAN's *eps* and *minPts*). If you hit bugs or surprising results, explain how you diagnosed and fixed them. Imagine you are walking a classmate through your solution and the reasoning behind each step. Keep this section concise yet informative.

Results and Analysis: Highlight the cluster count(s) you ultimately chose, which algorithm(s) performed best, and point out any samples or regions that remained ambiguously assigned or clearly mis-clustered. Discuss how each validation metric (Elbow inertia, Silhouette score, Calinski–Harabasz, Davies–Bouldin, dendrogram cutting, etc.) supported or contradicted your decision. Explain what these scores reveal about the intrinsic structure of the data e.g., whether clusters are well-separated, overlapping, or vary greatly in density. This section should show that you can interpret clustering diagnostics and relate them to data characteristics and algorithm behavior.

Use of AI: This is a *mandatory* disclosure of any AI assistance you used while completing Parts 1 and 2. Be specific about:

- **Which tools or platforms have you used?**
- **For what tasks did you use AI and how?** Did you use it to get hints on syntax, generate a snippet of code, debug an error message, or explain a concept you were unsure about?
- **How you integrated the AI's suggestions into your work.** It's important to clarify that you didn't just copy blindly.
- **Any instances where AI was wrong or not directly helpful.** It's valuable to mention if the AI gave incorrect advice that you caught or if you decided not to use an AI suggestion.
- **Citation of AI content**: If you have any code or text in your submission that was significantly generated by AI, cite it. You can do this informally in this report. Formal citation format is not required, but clarity is. If you used AI for brainstorming or minor phrasing, just acknowledge it generally. The key is transparency.

Remember, using AI is permitted in this assignment, but you must disclose it. We are grading you on your understanding and the work you did around the AI tools, not just on raw code. Unreported AI usage that is later detected will be treated as a violation of academic integrity. On the other hand, the well-documented use of AI (even if it contributed a lot) will be viewed as additional credit as you leverage resources effectively and ethically.

2.2 Report format:

- **Aim for a clear, well-organized report of about 1-5 pages**. Quality matters more than length. You can integrate your answers to the analysis questions into the narrative, but ensure all points are addressed.
- **The report can be submitted as a PDF document.** If you're writing in the Jupyter Notebook, you can compile your markdown answers into a PDF. Just make sure it's readable.
- **Be honest and specific when describing AI use.** This report is not only graded but also serves as a record of your ethical use of AI. Being truthful here will not negatively affect your grade; on the contrary, thorough documentation of AI use is part of the grading criteria.
- **Write in complete sentences and your own voice.** This is not a formal essay – it can be somewhat conversational – but it should be well-written and clear. Avoid just bulleting answers without context.

Grading and Evaluation Criteria

Your submission will be evaluated based on both correctness and your demonstrated understanding of the material. The assignment is worth a total of 100 points, distributed among the three parts as follows:

Grading Rubric Summary

- **Part 1 (15 pts):** 6 code completions (2.5 pts each).
  - Full: Code passes all tests.
  - Partial: Minor errors; concept is correct.
  - Zero: Blank or completely incorrect.
- **Part 2 Code (20 pts):**
  - Implementation / Model-building: 10 pts.
  - Metrics Calculation/ Validation: 5 pts.
  - Code Clarity and organization: 5 pts.
- **Part 2 Analysis (25 pts):**
  - Correct Cluster-Count Selection & Clear Clustering Plot: 5 pts
  - Discussion of Metric Agreement/Conflict: 5 pts
  - Representation Effect (raw vs PCA vs AE): 5 pts
  - Stability Assessment: 5 pts
  - Alternative Approach or Improvement: 5 pts
- **Part 2 Rigor (10 pts):** Range of experiments, reproducibility, and thoughtful iteration.
- **Part 3 Clarity (10 pts):** Well-organized, all required sections.
- **Part 3 Depth (10 pts):** Detailed reflection on process and learning.
- **Part 3 AI Documentation (10 pts):** Complete and honest disclosure of AI usage.
- **Bonus:** Up to 5 pts for exceptional work.

Part 1: Fill-in-the-Blanks Code (15 pts)

- Each coding blank or sub-task is allocated approximately 5 points. We will automatically grade this section by running your completed code against expected outputs.
- **Full Credit:**Code runs without error and produces the expected results (e.g., valid cluster labels for k-Means and DBSCAN, and a correctly populated inertia array for the Elbow plot).
- **Partial Credit:** Awarded if the logic is mostly correct but minor bugs or parameter issues remain; TAs will manually review such cases.
- **Evaluation Focus:**
  - Correct completion of the code as required.
  - Whether the code produces the expected result on sample tests.

Part 2: Coding Task & Analysis (55 pts)

This part is assessed both on the code you implement and the conceptual analysis you provide.

Code Implementation (20 pts)

- **Clustering Algorithms & Pre-processing (10 pts):**
  - Correct implementation and sensible parameter-tuning for every required method (k-Means, DBSCAN, Agglomerative, and the AutoEncoder used for dimensionality reduction).
  - Full credit if all algorithms converge and produce valid cluster labels; partial credit if any method is missing or contains non-fatal bugs.
- **Metric Computation & Visualisation (5 pts):**
  - Accurate calculation of internal validation metrics (Elbow inertia, Silhouette score) and correct generation of required plots (Elbow curve, Silhouette Score, dendrogram, PCA/t-SNE scatter, etc.).
- **Code Clarity and Adherence (5 pts):**
  - Use of methods taught in class (and avoidance of banned techniques).
  - Readability, proper commenting, and elimination of extraneous code. AI-generated code must be cleaned up and integrated correctly.

Results & Conceptual Analysis (25 pts)

- **Correct Cluster-Count Selection & Clear Plot (5 pts)**
  - Justify the chosen number of clusters (or DBSCAN parameters) and provide a well-labelled visualisation of the final clustering.
- **Metric Agreement/Conflict Discussion (5 pts)**
  - Analyse how Elbow/Silhouette (and, where relevant, dendrogram or DBSCAN heat-map) align or disagree and what that indicates about cluster structure.
- **Representation Effect (5 pts)**
  - Compare clustering quality across the *raw*, *PCA*, and *AutoEncoder* feature spaces. State which representation produced the clearest separation and why.
- **Stability Assessment (5 pts)**
  - Run one algorithm several times and report the variation in clustering performance. Comment on what this reveals about the reliability of the clustering.
- **Alternative Approach or Improvement (5 pts)**
  - Implement one concrete enhancement and briefly discuss the results.

Experimental Rigor & Iteration (10 pts)

- **Range of Experiments:**
  - Explored a broad spectrum of hyper-parameters to demonstrate metric stability e.g. tested k from 2 to 15 (or more) for k-Means / Agglomerative, several *eps*/*minPts* combinations for DBSCAN, and different linkage options for dendogram plot.
  - Full credit requires sufficient coverage to reveal how cluster quality changes across settings, not just a single "good" configuration.
- **Reproducibility:**
  - Used fixed random seeds (e.g. `random_state=42`) for algorithms that involve randomness (k-Means initialisation, train/test splits for AutoEncoder training, etc.).
- **Investigation of Anomalies:**
  - When metrics or visualisations behave unexpectedly (e.g. Silhouette score rising then abruptly dropping, many single-point "noise" clusters in DBSCAN), include a brief discussion or follow-up tests (e.g. change preprocessing techniques etc.) to diagnose the issue.

Part 3: Report & AI Documentation (30 pts)

- **Clarity and Organization (10 pts):**
  - The report should be well-structured with clear sections (approach, clustering findings, AI usage, and reflection).
  - Points may be deducted for disorganized writing or missing sections.
- **Depth of Reflection (10 pts):**
  - Provide a meaningful discussion of your problem-solving process, including challenges, decisions, and learning outcomes.
  - Superficial or generic reflections (e.g., "everything went well") will receive partial credit.
- **AI Usage Documentation (10 pts):**
  - Explicitly document all instances of AI assistance (e.g., naming the tools used, the specific help received, and how you integrated the responses).
  - Full credit for comprehensive and honest documentation; failure to mention AI usage when evidence exists will result in significant penalties.

Bonus: Quality of Insights (+ up to 5 pts)

- Extra points may be awarded for exceptionally innovative analysis, extra experiments that add value, or particularly thoughtful insights in your reflections.