Course Project: Dimensionality Reduction, and Hybrid Recommender Systems Implementation and Analysis

Discussion and Submission Deadline: Week 15, Monday, 5 January 2026 from 11:00 AM

Total Weight: 30% (SECTION 1: 10% and SECTION 2: 20%)

Using AI assistance (ChatGPT, Copilot, etc.) is allowed for learning but must be acknowledged

1. Project Instructions

1.1 General Guidelines

- This project document consists of 14 pages in total. Students must review and complete all requirements in the specified order.
- The project is divided into **TWO MAJOR SECTIONS**:
  - SECTION 1 (10%): Dimensionality Reduction and Matrix Factorization Methods (PCA and SVD).
  - SECTION 2 (20%): Design and implementation of a Complete Recommendation Engine

1.2 Team Formation and Registration

- Students must work in groups of **FOUR** students.
- Each team member must contribute equally and be responsible for specific components.
- Team Formation Email Deadline: Tuesday, December 30, 2025, at 11:59 PM.
- Email Requirements: Send to Lab TA (CC: Course Instructor) with:
  - Subject: AIE425 - Final Project Team Registration - [Team Leader ID and Name]
  - Body must include:
    - All team member names and IDs (as registered in SIS)
    - Assigned responsibilities for each member in SECTION 1 and SECTION 2:
      Note: All members must contribute to data analysis, modeling, report integration, quality check, and code submission.
    - Selected dataset for SECTION 1 (from Assignment 1 or new dataset meeting minimum requirements)
    - Selected application domain for SECTION 2 (from ITEM 4.2)

Group Number Assignment:

- Official Group Numbers will be announced on Thursday, January 1, 2026, via CANVAS.
- Please monitor CANVAS announcements closely.
- These assigned Group Numbers must be used in:
  - All file names and submissions
  - GitHub repository naming
  - The Week 15 discussion schedule
  - All project documentation
- Example: If assigned Group 5, use AIE425_FinalProject_Group5.zip

1.3 Discussion and Submission Requirements

Pre-Discussion Requirements, GitHub and CANVAS Submission:

- Students must use their GitHub accounts created in Assignment 1.
- All work must be prepared and uploaded to both GitHub and CANVAS **BEFORE** the start of Week 15 discussions.

Mandatory Pre-Discussion Review:

- Teams MUST schedule and complete a pre-discussion review of their submissions/deliverables on both GitHub repository and CANVAS with their Lab TA **BEFORE** Week 15 discussions begin.

  Submission Checklist
  - **GitHub Repository:**
    - Complete code for SECTION 1 and SECTION 2
    - All required folders and files per repository structure (ITEM 9.3)

- *README.md* with clear instructions
- *requirements.txt* with all dependencies
- Results (plots, tables, saved models)
- Repository is public or instructor/TAs added as collaborators

- **CANVAS Submission** (see ITEM 9.2)
  - ZIP file: AIE425_FinalProject_Group[X].zip
  - Final report (PDF + DOCX formats)
  - Plagiarism report (similarity ≤30%)
  - GitHub repository URL in submission comments
- **Pre-Discussion TA Review:**
  - Review appointment scheduled with Lab TA
  - Both GitHub and CANVAS submissions reviewed
  - TA confirmation obtained

- ***Failure to complete the pre-discussion review with the TA means your team will NOT be permitted to present during Week 15 discussions.***

Notes on the Report:

- Reports must be word-processed (HANDWRITTEN scans are NOT ACCEPTED).
- Follow formatting guidelines specified in (ITEM 6).

Week 15 Discussion Session

- Date and Time: Monday, January 5, 2026, starting at 11:00 AM.
- The detailed presentation schedule will be announced on Sunday, January 4, 2026 via CANVAS.
- Each team will be assigned 15-minute presentation (using slides) and live demonstration.
  - Live demonstration tested and ready.
  - All team members prepared to present and answer questions.
  - SECTION 1 overview and key findings (4-5 minutes)
  - SECTION 2 system demonstration and results (6-8 minutes)
  - Participate in Q&A session (technical depth assessment and individual contribution verification) with teaching team (2-5 minutes).

Attendance Requirement:

- ALL team members MUST attend and participate in the Week 15 discussion session.
- Each team member must be prepared to answer questions about their specific contributions.
- ***Failure to attend Week 15 discussion session will result in ZERO for this project.***

1.4 Academic Integrity

- Use of AI assistance (LLMs: ChatGPT, Copilot, Claude, etc.) is allowed for learning purposes and code optimization but MUST be explicitly documented in an "AI Assistance Appendix".
- All AI-assisted code or content must be clearly marked and attributed.
- Plagiarism detection will be performed; similarity >30% is unacceptable.

2. Introduction

This final course project blends all concepts covered in AIE425. You will demonstrate mastery of neighborhood-based collaborative filtering, matrix factorization techniques, and content-based recommendation through comprehensive implementation, analysis, and evaluation.

SECTION 1 (10%) extends Assignment 1 by implementing PCA with Mean-Filling and Maximum Likelihood Estimation, Singular Value Decomposition (SVD) for collaborative filtering, Latent Factor Models and comprehensive comparative analysis. This section consists of THREE parts:

Part 1: PCA Method with Mean-Filling.
Part 2: PCA Method with Maximum Likelihood Estimation.
Part 3: Singular Value Decomposition (SVD) method

SECTION 2 (20%) challenges you to design, implement, and deploy a complete domain-specific recommender system that integrates neighborhood CF methods, matrix factorization and modern content-based filtering, implements hybrid recommendation strategies, addresses real-world challenges such as cold-start, sparsity, and scalability, and demonstrates thorough evaluation using multiple metrics. This section consists of THREE parts:

Part 1: Domain Analysis and Data Preparation
Part 2: Content-Based Recommendation
Part 3: Collaborative Filtering and Hybrid Approach

3. SECTION 1: Dimensionality Reduction and Matrix Factorization (10%)

Round all numerical values to 2 decimal places unless otherwise specified.

3.1. Dataset Requirements

1. Use the same dataset from Assignment 1, OR
2. Select a new dataset meeting these minimum requirements:

- ≥ 10,000 users
- ≥ 500 items
- ≥ 100,000 interactions/ratings
- Ratings on 1-5 scale (adjust if necessary)

If using a new dataset:

- Perform complete statistical analysis as in Assignment 1, Section ONE
- Select target users (U1, U2, U3) and target items (I1, I2) following Assignment 1 criteria
- Cold user (≤2% ratings)
- Medium user (2% ≤ ratings ≤ 5%)
- Rich user (>10% ratings)
- Low popularity, medium popularity, high popularity items
- Save all preprocessing results for use throughout Part 1

3.2. Part 1: PCA Method with Mean-Filling

Use the PCA method with mean-filling technique and compute the covariance matrix then compute the rating prediction for the target items I1 and I2.

1- Calculate the average rating for each of the target items (I1 and I2).
2- Use the mean-filling method to replace the unspecified ratings of each of the target items (I1 and I2) with its corresponding mean value.
3- Calculate the average rating for each item.
4- For each item, calculate the difference between ratings and the mean rating of the item.
5- Compute the covariance for each two items.
6- Generate the covariance matrix.
7- Determine the top 5-peers and top 10-peers for each of the target items (I1 and I2) using the transformed representation (covariance matrix).
8- Determine reduced dimensional space for each user in case of using the top 5-peers.
9- Use the results from point 8 compute the rating predictions of the original missing rating for each of the target items (I1 and I2) using the top 5-peers.
10- Determine reduced dimensional space for each user in case of using the top 10-peers.
11- Use the results from point 10 to compute the rating predictions of the original missing rating for each of the target items (I1 and I2) using the top 10-peers.
12- Compare the results of point 9 with results of point 11. Comment on your answer.

3.3. Part 2: PCA Method with Maximum Likelihood Estimation

Use the PCA method with MLE technique and compute the covariance matrix then compute the rating prediction for the target items I1 and I2. For simplicity, assume the Maximum Likelihood Estimate of the covariance between each pair of items is estimated as the covariance between only the specified entries. i.e, only the users that have specified ratings for a particular pair of items are used to estimate the covariance. If there are no users in common between a pair of items, the covariance is estimated to be 0.

1. Generate the covariance matrix.
2. Determine the top 5-peers and top 10-peers for each of the target items (I1 and I2) using the transformed representation (covariance matrix).
3. Determine reduced dimensional space for each user in case of using the top 5-peers.
4. Use the results from point 3 compute the rating predictions of the original missing rating for each of the target items (I1 and I2) using the top 5-peers.
5. Determine reduced dimensional space for each user in case of using the top 10-peers.
6. Use the results from point 5 to compute the rating predictions of the original missing rating for each of the target items (I1 and I2) using the top 10-peers.
7. Compare the results of point 3 with results of point 6. Comment on your answer.
8. Compare the results of point 9 in part 1 with results of point 4. Comment on your answer.
9. Compare the results of point 11 in part 1 with results of point 6. Comment on your answer.

Discussion and Conclusion:

prepare a report that includes the following

1. A section titled "Outcomes" summarizing the key findings and results from all parts and case studies.
2. A section called "Summary and Comparison" summarizes and compares the results of part 1 and part 2, emphasizing the accuracy of predicting the missing rating, and the pros and cons of each method.
3. A "conclusion" section, which summarizes your own comments and conclusion, shows the impact of Maximum Likelihood Estimation.

3.4. Part 3: Singular Value Decomposition (SVD) for Collaborative Filtering

SVD generalizes eigenvalue decomposition to non-square matrices, decomposing the ratings matrix R into a product of three matrices, i.e. $R = U\Sigma V^T$, where U that is $m \times k$ (*user-feature matrix*), V that is $k \times n$ (*item-feature matrix*), $\sum$ is $k \times n$ diagonal (*singular values*), and k is the number of latent factors (*reduced dimensionality*). In this PART, apply both SVD and the truncated SVD on the ratings matrix (by adopting the low-rank assumption,) to approximate your full dimensional matrix using a small number of factors (*k-features*) that capture the data's essential structure and main patterns.

Tasks and questions

1. Data Preparation

1.1. Load your ratings matrix from Assignment 1 or your new preprocessed dataset.
1.2. Calculate the average rating for each item ($\bar{r}_i$).
1.3. Apply mean-filling: replace missing ratings with the item's average rating.
1.4. Verify matrix completeness (no missing values).

2. Full SVD Decomposition

2.1. Compute the full SVD: $R = U\Sigma V^T$
2.2. calculate and save:

- Eigenpairs (λ_i , v_i)
- All singular values $(\sigma_1, \sigma_2, ..., \sigma_n)$ $\rightarrow$ build Σ

- Normalize $v_i \to e_i = \frac{v_i}{\|v_i\|} \to$ orthonormal vectors (columns of V) $\to V^T$
- Calculate $u_\mathrm{i} = \frac{\mathrm{A} \cdot \mathrm{e_i}}{\sigma_\mathrm{i}} \to$ build corresponding vectors (columns of U)

2.3. Verify orthogonality:
- Check $U^T U = 1$and $V^T V = 1$ (identity matrix)
- Report any deviations from orthogonality

2.4. Visualize:
- Plot singular values in descending order
- Create scree plot showing variance explained by each singular value

3. Truncated SVD (Low-Rank Approximation)

3.1. Implement truncated SVD for different values of k (*latent factors*):
- $k = 5, 20, 50, 100$

3.2. For each k value:
- Construct U_k (first k columns of U)
- Construct Σ_k (top-left $k \times k$ submatrix of Σ)
- Construct V_k (first k columns of V)
- Compute approximation: $\hat{R}_k = U_k \cdot \Sigma_k \cdot V_k^T$

3.3. Calculate reconstruction error for each k:
- Mean Absolute Error (MAE) on all ratings
- Root Mean Square Error (RMSE) on all ratings

3.4. Create visualizations:
- Plot reconstruction error vs. k (elbow curve for SVD)
- Plot percentage of variance retained vs. k
- Identify optimal k using elbow method

4. Rating Prediction with Truncated SVD

4.1. For each optimal k value (from elbow analysis):

4.2. Predict missing ratings for target items (I1, I2):
- For each target user (U1, U2, U3)
- Extract user's latent factor representation from U_k
- Extract item's latent factor representation from V_k
- Compute predicted rating: $\hat{r}_{ui} = u_k^T \cdot \Sigma_k \cdot v_k$

4.3. Record all predictions in a structured table.

4.4. If ground truth is available for target items:
- Calculate prediction accuracy (MAE, RMSE)
- Compare with Assignment 1 predictions

5. Comparative Analysis: SVD vs. PCA Methods

5.1. Compare reconstruction quality:
- SVD (this part) vs. PCA with mean-filling (part 1)
- SVD (this part) vs. PCA with MLE (part 1)

5.2. Compare prediction accuracy:
- Rating predictions for target items using all three methods

5.3. Compare computational efficiency:
- Time complexity analysis for each method
- Actual runtime measurements for:
  - Matrix decomposition
  - Rating prediction
- Memory requirements

5.4. Create comparison tables showing:
- Reconstruction errors

- Prediction errors (MAE, RMSE)
- Runtime (seconds)
- Memory usage (MB)

6. Latent Factor Interpretation

6.1. Analyze the top-3 latent factors (largest singular values):

6.2. For each latent factor:

- Identify items with highest absolute values in V
- Identify users with highest absolute values in U
- Attempt to interpret the semantic meaning of each factor
- Provide examples (e.g., "Factor 1 may represent action movies preference")

6.3. Visualize latent space:

- Project users and items onto the first 2 latent factors
- Create scatter plot showing user-item relationships
- Color-code by user activity level or item popularity

7. Sensitivity Analysis

7.1. Test robustness to missing data:

- Vary the percentage of missing ratings (10%, 30%, 50%, 70%)
- For each percentage, perform SVD and measure:
  - Reconstruction error
  - Prediction accuracy
- Plot error vs. missingness percentage

7.2. Test impact of initialization:

- Try different mean-filling strategies:
  - Item mean vs User mean
- Compare resulting predictions

8. Cold-Start Analysis with SVD

8.1. Simulate cold-start users (users with ≤ 5 ratings):

- Randomly select 50 users with > 20 ratings
- Hide 80% of their ratings to create cold-start scenario

8.2. For each cold-start user:

- Estimate user latent factors using limited ratings
- Predict ratings for unrated items
- Compare with ground truth (hidden ratings)

8.3. Evaluate cold-start performance:

- Calculate MAE, RMSE for cold-start users
- Compare with warm-start users (full rating history)
- At what point (number of ratings) does performance become acceptable?

8.4. Propose and test cold-start mitigation strategies:

- Hybrid approach (combine SVD with item popularity)
- Content-based initialization of latent factors
- Measure improvement from baseline

9. Discussion and Conclusion for PART 3

Prepare a comprehensive report section including:

1. Summary of Findings

- Key results from SVD analysis
- Optimal number of latent factors (k) and justification
- Performance comparison: SVD vs. PCA methods

2. Method Comparison Table

Create a detailed table comparing:

- Reconstruction error
- Prediction accuracy (MAE, RMSE)
- Time and Space complexities (theoretical and measured)
- Handling of sparsity
- Cold-start performance

3. Critical Evaluation

- Strengths and weaknesses of each method
- When to use SVD vs. PCA with mean-filling vs. PCA with MLE
- Impact of dataset characteristics on method choice

4. Lessons Learned

- Challenges encountered during implementation
- Solutions applied
- Insights gained about matrix factorization

4. SECTION 2: Design and implementation of a Complete Recommendation Engine (20%)

4.1. Overview

Groups of **FOUR students** will design and implement a complete recommendation engine for a specific application domain. Your submission must include:

- System description and architecture
- Data collection and preprocessing
- Implementation of content-based and hybrid recommendation approaches
- Evaluation with multiple metrics

4.2. Domain Selection

Each team must select ONE application domain from the list below.

Selection Deadline: Tuesday, December 30, 2025, at 11:59 PM.

Email your selection to Lab TA (CC: Course Instructor) as specified in ITEM 1.2.

Available Application Domains

1. Custom Domain - Propose your own innovative recommendation domain (submit detailed description by Tuesday, December 30, 2025)
2. Generative AI Prompt Recommendation Engine (optimized prompts based on task categories and success rates)
3. Multi-Modal Product Recommendation Engine (combining images, text, and user behavior)
4. Sustainable Fashion and Circular Economy Engine (second-hand/eco-friendly apparel with carbon footprint scores)
5. Podcast Recommendation Engine with Transcript Analysis
6. Short-Form Video Content Recommendation (dwell time and visual/tag-based features)
7. Live Streaming Content Recommendation Engine
8. Music Playlist Generation with Mood Detection
9. Adaptive Learning Path Recommendation with Prerequisite Mapping
10. Personalized Learning Path Recommendation (video modules based on skill gap analysis)
11. Research Paper and Citation Recommendation for Students
12. Programming Language Learning Content Recommendation (adaptive to proficiency level)
13. AI-Powered Mental Wellness and Meditation Guide (audio/breathing exercises based on mood patterns)
14. Mental Health Resource Recommendation with Privacy Preservation
15. Sleep Quality Improvement Recommendation Engine
16. Telemedicine Provider Recommendation Engine

17. Remote Job Matching Recommendation Engine
18. Professional Networking and Collaboration Recommendation
19. Skill Development and Certification Recommendation
20. Freelance Project Recommendation for Gig Workers
21. Community Event Recommendation with Location Intelligence
22. Volunteer Opportunity Matching Engine
23. Local Business Discovery and Recommendation
24. Interest-Based Group Formation Recommendation
25. Robo-Advisor Investment Recommendation with Risk Profiling
26. Blockchain Cryptocurrency and Non-Fungible Tokens (NFTs) Recommendation Engine
27. Micro-Investment Opportunity Recommendation
28. Financial Literacy Content Recommendation
29. Peer-to-Peer Lending Opportunity Recommendation
30. Public Transportation Route Recommendation
31. Energy Consumption Optimization Recommendation
32. Urban Green Space and Recreation Recommendation
33. Carbon Footprint Reduction Action Recommendation
34. Recycling and Waste Management Recommendation
35. Renewable Energy Solution Recommendation for Households
36. Sustainable Agriculture Practice Recommendation
37. Research Collaboration Recommendation (co-author matching)
38. Patent and Prior Art Recommendation for Inventors
39. Grant and Funding Opportunity Recommendation
40. Lab Equipment and Resource Sharing Recommendation
41. Open Dataset Recommendation for Researchers
42. Subscription-Based Micro-SaaS Tool Finder (software recommendations based on tech-stack compatibility)

4.3. Requirements for SECTION 2

Part 1: Domain Analysis and Data Preparation (4%)

1. Domain Background and Requirements

1.1. Provide brief domain background (1-2 paragraphs):
Current recommendation approaches in this domain, your proposed system focus, and Target users

1.2. Identify key domain challenges (select 2-3):
Cold-start scenarios, data sparsity, real-time requirements.

2. Dataset Preparation

2.1. Data source:
Use publicly available dataset OR create synthetic data
Minimum requirements: 5,000 users, 500 items. 50,000 interactions/ratings. Content features (text descriptions, metadata, or categories)

2.2. Data preprocessing:
Handle missing values and duplicates and scale ratings to 1-5 range.
Extract basic statistics: Number of users, items, ratings. Sparsity level and rating distribution

2.3. Basic Exploratory Analysis:
Plot user activity and item popularity distribution. Identify if long-tail problem exists

Part 2: Content-Based Recommendation (8%)

3. Feature Extraction and Vector Space Model

3.1. Text feature extraction (choose ONE approach):
- TF-IDF vectors with basic preprocessing (tokenization, stop-word removal)
- OR Bag-of-Words with term frequency

3.2. Additional features (if applicable to your domain):
- Categorical features (genre, category, ..). Numerical features (price, duration, rating)

3.3. Create item-feature matrix and document your feature selection.

4. User Profile Construction

4.1. Build user profiles:
- Weighted average of rated item features (weight by rating value)
- OR Simple average of item features for items rated ≥ 4

4.2. Handle cold-start users (choose ONE strategy):
- Use popular item features
- OR Demographic-based initialization (if demographic data available)

5. Similarity Computation and Recommendation

5.1. Compute similarity:
- Use Cosine similarity between user profiles and all items
- Create user-item similarity scores

5.2. Generate top-N recommendations:
- Rank items by similarity score, remove already-rated items.
- Return top-10 and top-20 recommendations

6. k-Nearest Neighbors (k-NN)

6.1. Implement item-based k-NN:
- Find k most similar items for each item (k = 10, 20)
- Predict ratings using weighted average of similar items

6.2. Compare content-based and k-NN approaches.

7. Complete Numerical Example

7.1. Provide step-by-step example showing:
- Sample item descriptions
- TF-IDF calculation for 3-5 sample items
- User profile from 3-5 ratings
- Similarity scores
- Top-5 recommendations with scores

Part 3: Collaborative Filtering and Hybrid Approach (8%)

8. Collaborative Filtering Integration

8.1. Implement ONE CF approach:
- User-based OR Item-based CF (using techniques from Assignment 1)
- Use cosine similarity or Pearson correlation

8.2. Use matrix factorization from SECTION 1:
- Apply SVD with k=10 or k=20 latent factors
- Generate predictions for target users

9. Hybrid Recommendation Strategy

9.1. Implement ONE hybrid approach:
- ***Option A:*** Weighted Hybrid:
  - Combine content-based and CF scores: Score $= \alpha \times CB + (1-\alpha) \times CF$
  - Test $\alpha = 0.3, 0.5, 0.7$
  - Select best α based on validation performance

Option B: Switching Hybrid:
Use CF for users with ≥ 10 ratings, and content-based for users with <10 ratings
Option C: Cascade Hybrid:
Content-based to filter top-50 candidates, and CF to rank final top-10

9.2. Justify your choice based on domain characteristics.

10. Cold-Start Handling

10.1. Demonstrate cold-start solution:
Test on users with 3, 5, and 10 ratings
Show how your hybrid approach handles limited data
Compare with popularity baseline

11. Baseline Comparison

11.1. Compare your hybrid system against:
Random recommendations, most popular items, and pure content-based

11.2. Create comparison table showing all metrics.

12. Results Analysis

Which approach was performed best?
How well does hybrid handle cold-start?

4.4. Implementation Requirements

1. Code Structure

Required files:
- data_preprocessing.py (data loading and cleaning)
- content_based.py (TF-IDF, user profiles, similarity)
- collaborative.py (CF implementation)
- hybrid.py (hybrid approach)
- evaluation.py (metrics calculation if any - bonus)
- main.py (run complete pipeline)
- README.md (instructions to run code)
- requirements.txt (dependencies)

2. Code Quality

- Clear comments for major steps, Function docstrings, Reproducible (set random seeds)

3. Visualizations (minimum required)

- Rating distribution histogram and user/item activity distribution
- Comparison bar chart (metrics across methods)
- Top-10 recommendation example for 2-3 sample users

4.5. Report Requirements for SECTION 2

Your report for SECTION 2 should include (6-9 pages):

1. Introduction (1 page)
Domain description, system objectives, and key challenges addressed
2. Data and Methodology (1-2 pages)
Dataset description and statistics. Feature extraction approach
Content-based method, collaborative filtering approach, hybrid strategy
3. Implementation (2-3 pages)
System architecture diagram, key implementation decisions, complete numerical example
4. Evaluation and Results (1-2 pages)
Evaluation methodology, metrics comparison table, results analysis and discussion
5. Discussion and Conclusion (1 page)
What worked well, limitations, domain-specific insights, and lessons learned
6. Appendices
Appendix A: Sample code snippets. Appendix B: Additional visualizations

5. Coursework Assessment Criteria

- Weighting: This coursework contributes **30%** to the overall course grade and must be completed in GROUPS OF FOUR.
- Purpose: The task is designed to evaluate the progression of the student's academic capabilities and to assess the achievement of the course's Intended Learning Outcomes (ILOs).
- Assessment Criteria: Marks will be allocated proportionally based on the following aspects:
    - Depth and accuracy of the technical implementation.
    - Evidence of academic rigor and critical thinking.
    - Clarity and organization of information.
    - Quality and professionalism of the written discussion.
- Penalty for Lack of Thoroughness: If a submission lacks sufficient evidence of thorough analysis and methodology, a **40%** mark reduction will be applied.
- Partial Credit for Near-Miss Attempts: In cases where the student's approach is conceptually sound but contains errors, the maximum score awarded for that section will be capped at **70%**.

6. Written Report (paper)

6.1. Report content

- Cover Page should include:
    - Line 1: AIE425 Intelligent Recommender Systems, Fall Semester 25/26
    - Line 2: Final Course Project
    - Line 3: Group [X] - Team Member Names and IDs
    - Line 4: Submission Date: Monday, January 5, 2026
- Report Sections:
    - Executive summary (1-2 pages maximum)
    - SECTION 1: Dimensionality Reduction and Matrix Factorization
      Complete solutions step-by-step, tables, visualizations, and analysis
    - SECTION 2: Domain-Specific Recommender System
      System description, implementation, results, and analysis
    - Overall Conclusions
    - References (if any external sources used)
    - Appendices:
      Appendix A: AI Assistance Acknowledgment
      Appendix B: Team Contribution Breakdown
      Appendix C: Additional Visualizations
    - File naming: AIE425_FinalProject_Group[X]_Report

6.2. Report Format:

Your report must follow these formatting standards:

- Must be typed and follow a professional report format and on A4-sized paper.
- Use correct grammar, formal language, appropriate tenses, and standard English spelling.
- The cover page must have double line spacing, be center-aligned, and not numbered.
- All other pages:
    - Use 1.5 line spacing.
    - Use Arial 12 pt font for body text and bold Arial 12 pt for section headings.
    - Page numbers must appear centered at the bottom in the format: Page X of Y.
    - Headings must maintain uniform font, size, and alignment.

7. Plagiarism and Academic Honesty

- AI Assistance: Use of AI tools (ChatGPT, Copilot, Claude, etc.) is allowed for learning and code optimization but MUST be acknowledged in "Appendix A: AI Assistance Acknowledgment".
- Integrity and Collaboration: Teams of FOUR must work together. All team members must contribute to data analysis, modeling, report integration, and code submission.

- Copying code from online sources without attribution is strictly prohibited and may result in course failure.
- Submit your team's original work. If you use content not created by your team, it must be explicitly cited and properly referenced.
- Code plagiarism from other groups will result in ZERO for all involved parties.
- External libraries (scikit-learn, NumPy, pandas, Surprise, etc.) are allowed and encouraged. Properly cite any code snippets from online sources.
- A Plagiarism Report is mandatory with submission. Submissions exceeding 30% similarity will be rejected.

8. Feedback given to students in response to assessed work.

- During the Week 15 discussion session, teams will receive oral feedback on their presentation and implementation.
- If students need additional input, they are encouraged to speak with the teaching staff.

9. Deliverables and Submission Instructions

9.1 Submission Deadlines and Requirements

Pre-Submission Deadline: Before your discussion session
- Submit complete package to both CANVAS and GitHub
- Schedule and complete pre-discussion review with Lab TA

Discussion Session: Monday, January 5, 2026, 11:00 AM
- All team members must attend, 15-minute presentation + Q&A

9.2 CANVAS Submission: Submit a ZIP file containing

- Written report (PDF or DOCX format)
- Complete source code
- Dataset files (or link to dataset in README.md if too large)
- README.md and requirements.txt
- Plagiarism Report (similarity ≤30%)
- File naming convention: AIE425_FinalProject_Group[X].zip

9.3 GitHub Submission:

Repository Setup

- Create a public repository named: AIE425_FinalProject_Group[X]
- **Push** all code, documentation, and results to GitHub
- Submit the GitHub repository URL on CANVAS in the submission comments
- Ensure the repository is accessible (add instructor and TA as collaborators)
- Repository Structure:

```
AIE425_FinalProject_Group[X]/
├── README.md
├── requirements.txt
├── PLAGIARISM_REPORT.pdf
├── SECTION1_DimensionalityReduction/
│   ├── data/
│   ├── code/
│   │   ├── pca_mean_filling.py
│   │   ├── pca_mle.py
│   │   ├── svd_analysis.py
│   │   └── utils.py
│   ├── results/
│   │   ├── plots/
```

```
│    │    └── tables/
│    └── README_SECTION1.md
├── SECTION2_DomainRecommender/
│    ├── data/
│    ├── code/
│    │    ├── data_preprocessing.py
│    │    ├── content_based.py
│    │    ├── collaborative.py
│    │    ├── hybrid.py
│    │    └── main.py
│    ├── results/
│    └── README_SECTION2.md
└── Final_Report.pdf (or .docx)
```

Code requirements:

Well-documented Python code (.py or .ipynb); Clear comments explaining each major step; Function docstrings; Modular design (separate files for each component); Reproducible results; All code must be runnable in Visual Studio Code without modifications.

Use relative file paths only (no absolute paths like C:/Users/...).

No proprietary or platform-specific code.

9.4 File Naming and Identification

- All file paths in your code must be relative to the code file location.
- Group Number and team member names must appear at the top of each submitted file, including code files.
- Follow consistent naming: AIE425_FinalProject_Group[X]_[ComponentName]

9.5 Late Submission Policy:

- Within 24 hours with valid justification and prior arrangement: allowed, but with a **50%** grade deduction.
- **Beyond 24 hours NOT accepted**, and the project will receive a grade of **ZERO**.

10. Assessment Breakdown (30% Total)

10.1. SECTION 1: Dimensionality Reduction and Matrix Factorization (10%):

Component	Weight	Description
PCA Mean-Filling Implementation	3%	Covariance matrix, predictions, comparisons
PCA MLE Implementation	2%	MLE covariance, predictions, comparisons
SVD Implementation and Analysis	3%	Full/truncated SVD, latent factors, cold-start
Comparative Analysis	1%	Method comparison, complexity analysis
Code Quality and Report	1%	Documentation, clarity, professionalism
TOTAL SECTION 1	10%	

10.2. SECTION 2: Domain-Specific Recommender System (20%)

Component	Weight	Description
Domain Analysis and Data Preparation	4%	Dataset selection, preprocessing, statistical analysis
Content-Based Implementation	8%	TF-IDF, user profiles, k-NN, numerical example
Hybrid Approach and CF Integration	8%	CF implementation, hybrid strategy, cold-start, evaluation metrics, and results analysis
TOTAL SECTION 2	20%	

10.3. Presentation and Submission Quality

Component	Weight	Description
Week 15 Presentation	Mandatory	15-min presentation + Q&A; Failure to attend = -50% penalty
GitHub Repository Quality	Included	Proper structure, documentation, completeness
Plagiarism Report	Mandatory	Similarity ≤30%; >30% = potential ZERO

10.4. Overall Grading Rubric

Excellent (90-100%):
- Flawless implementation of all requirements
- Insightful analysis and critical evaluation
- Professional-quality report and well-documented code
- Novel contributions beyond requirements

Good (80-89%):
- Correct implementation with minor issues
- Solid analysis and evaluation
- Professional report and documented code
- Meets all requirements

Satisfactory (70-79%):
- Implementation with some errors but conceptually sound
- Adequate analysis with some missing depth
- Acceptable report quality
- Meets most requirements

Needs Improvement (<70%):
- Significant implementation errors
- Superficial analysis
- Poor report quality or incomplete deliverables
- Missing major requirements

-------------- END --------------