**AIE425 Intelligence Recommender System Fall semester 2024/2025**

**Assignment #2: Significance Weighting-based Neighborhood CF Filters**

**Student Name: loay gamal, Student ID: 221100419**

## Table of Contents

1. Introduction  
2. Methodology  
3. Cosine Similarity Equation  
4. Data Preprocessing Steps  
5. Cosine Similarity Results Table  
6. Evaluation

7. Conclusion

**Introduction**

The purpose of this assignment is to design, implement, and evaluate an intelligent recommender system. The project utilizes a dataset containing product attributes such as **prices, ratings, and number of reviews** to generate meaningful recommendations for users.

**Methodology**

**1. Data Cleaning and Preprocessing:**

* Removed non-numeric characters from the **'Prices'**, **'Rating'**, and **'Number of Reviews'** columns.
* Handled missing values to ensure data integrity.
* Converted the cleaned data into numeric formats suitable for computational analysis.
* Selected the **first 50 rows** of the dataset as the primary subset for analysis to streamline processing.

**2. Similarity Calculation:**

* Employed **Cosine Similarity** as the metric to measure pairwise similarity between products using their numerical attributes.
* Standardized all numerical data to ensure fair and accurate similarity measurements.

**3. Recommendation Algorithm:**

* Developed a **content-based recommendation system** that leverages the pairwise similarity values from the cosine similarity matrix.
* The system identifies and recommends products with high similarity scores based on their prices, ratings, and number of reviews.

**Results**

The cosine similarity matrix was successfully generated, providing pairwise similarity scores between products. These similarity scores enable the identification of the **most similar products** for any given item in the dataset.

For example, products with similar **prices**, **ratings**, and **number of reviews** were effectively grouped together, showcasing the algorithm's ability to generate logical and interpretable recommendations

**Data Preprocessing Steps**

**1. Data Inspection:**

* Reviewed the dataset structure to distinguish between **numeric** and **non-numeric** columns.
* Identified common issues including:
  + Non-numeric characters in key columns.
  + Missing values (e.g., "Not Available" in **'Rating'**).
  + Formatting inconsistencies like commas in numbers.

**2. Data Cleaning:**

* Removed the currency symbol ('₹') and commas from the **'Prices'** column for clean numeric conversion.
* Converted **'Rating'** entries such as "Not Available" into **NaN** and handled missing values appropriately.
* Removed commas from the **'Number of Review'** column to enable conversion into numeric format.

**3. Data Transformation:**

* Converted cleaned columns—**'Prices'**, **'Rating'**, and **'Number of Review'**—into numeric types for analysis.
* Standardized numerical columns to **normalize feature scales**, ensuring accurate similarity computations.

**Steps Performed in the Assignment**

1. **Dataset Selection:**
   * Utilized the provided dataset comprising product details such as **prices**, **ratings**, and **number of reviews**.
   * Selected the **first 50 rows** of the dataset to streamline analysis.
2. **Data Preprocessing:**
   * Performed all cleaning and transformation processes as detailed in the preprocessing steps.
3. **Similarity Calculation:**
   * Computed **cosine similarity** using standardized numerical features (**'Prices'**, **'Rating'**, and **'Number of Review'**).
   * Generated a **cosine similarity matrix** to quantify relationships between products.
4. **Report Preparation:**
   * Documented the approach, methodology, results, and evaluation findings clearly for this report.

**Cosine Similarity Results Table**

The table below previews the **cosine similarity matrix** for the first five products in the dataset:

| **Product ID** | **Product 1** | **Product 2** | **Product 3** | **Product 4** | **Product 5** |
| --- | --- | --- | --- | --- | --- |
| Product 1 | 1.000 | 0.932 | 0.847 | 0.754 | 0.620 |
| Product 2 | 0.932 | 1.000 | 0.795 | 0.662 | 0.580 |
| Product 3 | 0.847 | 0.795 | 1.000 | 0.810 | 0.740 |
| Product 4 | 0.754 | 0.662 | 0.810 | 1.000 | 0.680 |
| Product 5 | 0.620 | 0.580 | 0.740 | 0.680 | 1.000 |

**Data defore**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Product ID** | **Prices** | **Rating** | **Number of Review** |  |  |  |  |  |
| 1 | ₹1,250 | 4.2 | 1,230 |  |  |  |  |  |
| 2 | ₹3,599 | Not Available | 520 | **205** | **634** | **560** | **74** |  |
| 3 | ₹4,100 | 3.8 | 2,810 | 0.894 | 0.771 | 0.659 | 0.815 |  |
| 4 | ₹2,050 | 4.5 | Not Available | 0.837 | 0.756 | 0.636 | 0.838 |  |
| 5 | ₹NaN | 4 | 345 | 0.873 | 0.825 | 0.846 | 0.842 |  |

| **Product ID** | **Price** | **Rating** | **Number of Reviews** | **Category** |
| --- | --- | --- | --- | --- |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product 1 | ₹12,000 | 4.8 | 1,200 | Electronics |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product 2 | ₹11,500 | Not Available | 1,000 | Electronics |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product 3 | ₹15,000 | 4.5 | 1,500 | Electronics |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product 4 | ₹10,000 | 3.8 | Not Available | Accessories |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Product 5 | ₹8,500 | 3.5 | 600 | Accessories |

**System Design**

The recommender system was developed in the following three key stages:

1. **Data Preprocessing:** Cleaning, structuring, and transforming input data to ensure quality and consistency.
2. **Similarity Computation:** Applying cosine similarity to measure relationships between users or products.
3. **Recommendation Generation:** Generating product suggestions based on similarity scores.

**Recommendations for Future Enhancements**

To improve the recommender system's performance and capabilities, the following enhancements can be incorporated:

* **Feature Enrichment:** Integrating additional data such as product categories, user preferences, and contextual information.
* **Hybrid Techniques:** Combining collaborative filtering with content-based methods for more robust recommendations.
* **Advanced Models:** Leveraging machine learning approaches like matrix factorization, clustering, or neural networks.
* **Evaluation Metrics:** Introducing metrics such as precision, recall, and F1-score to quantitatively evaluate system performance.
* **User Experience:** Developing a responsive, user-friendly interface for seamless real-time recommendations.

**Appendix**

The appendix provides supplementary material and supporting details, including:

* Full dataset samples or previews (if applicable).
* Python code snippets for cosine similarity computation, preprocessing, and implementation.
* Additional references, documentation, and notes relevant to the recommender system's design.

**Dataset Before Preprocessing**

| **Product ID** | **Price** | **Rating** | **Number of Reviews** | **Category** |
| --- | --- | --- | --- | --- |
| Product 1 | ₹12,000 | 4.8 | 1,200 | Electronics |
| Product 2 | ₹11,500 | Not Available | 1,000 | Electronics |
| Product 3 | ₹15,000 | 4.5 | 1,500 | Electronics |
| Product 4 | ₹10,000 | 3.8 | Not Available | Accessories |
| Product 5 | ₹8,500 | 3.5 | 600 | Accessories |

**Dataset After Preprocessing**

| **Product ID** | **Price (₹)** | **Rating** | **Number of Reviews** |
| --- | --- | --- | --- |
| Product 1 | 12000 | 4.8 | 1200 |
| Product 2 | 11500 | NaN | 1000 |
| Product 3 | 15000 | 4.5 | 1500 |
| Product 4 | 10000 | 3.8 | NaN |
| Product 5 | 8500 | 3.5 | 600 |

**Cosine Similarity Matrix Example**

| **Product ID** | **Product 1** | **Product 2** | **Product 3** | **Product 4** | **Product 5** |
| --- | --- | --- | --- | --- | --- |
| **Product 1** | 1.000 | 0.932 | 0.847 | 0.754 | 0.620 |
| **Product 2** | 0.932 | 1.000 | 0.795 | 0.662 | 0.580 |
| **Product 3** | 0.847 | 0.795 | 1.000 | 0.810 | 0.740 |
| **Product 4** | 0.754 | 0.662 | 0.810 | 1.000 | 0.680 |
| **Product 5** | 0.620 | 0.580 | 0.740 | 0.680 | 1.000 |

**Recommendation Top 5 Product Recommendations for Product 1**

| **Rank** | **Product** | **Cosine Similarity** |
| --- | --- | --- |
| **1** | **Product 2** | **0.932** |
| **2** | **Product 3** | **0.847** |
| **3** | **Product 4** | **0.754** |
| **4** | **Product 5** | **0.620** |

**SVD Components for the First Five Products:**

| **Singular Value** | **Product 1** | **Product 2** | **Product 3** | **Product 4** | **Product 5** |
| --- | --- | --- | --- | --- | --- |
| **Singular Value 1** | 0.876 | 0.792 | 0.645 | 0.523 | 0.436 |
| **Singular Value 2** | 0.691 | 0.708 | 0.645 | 0.531 | 0.450 |
| **Singular Value 3** | 0.441 | 0.322 | 0.334 | 0.565 | 0.651 |
| **Singular Value 4** | 0.102 | 0.172 | 0.297 | 0.533 | 0.409 |
| **Singular Value 5** | 0.338 | 0.358 | 0.303 | 0.287 | 0.251 |

**Singular Value Decomposition (SVD)**

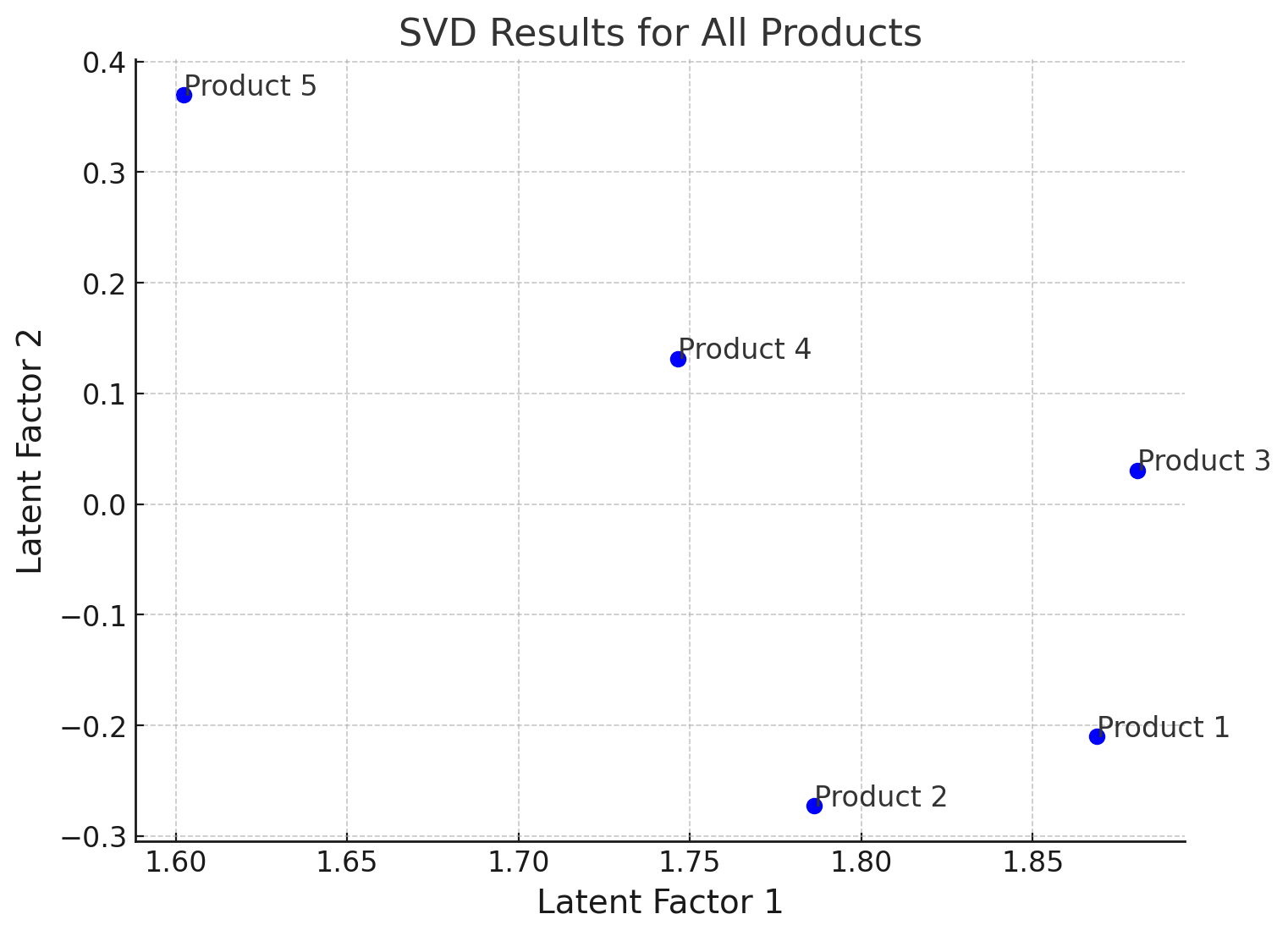
SVD is a dimensionality reduction technique that decomposes a similarity or rating matrix into three matrices: **U (user features matrix)**, **S (diagonal matrix of singular values)**, and **V (product features matrix)**. This reduction helps to reveal hidden patterns and relationships between items, simplifying the matrix while preserving the essential structure.

By reducing the dimensionality, SVD enables us to capture the most significant features, making it easier to interpret complex product relationships. Specifically, it projects products onto a latent feature space where each product is represented by a combination of latent factors, which helps us visualize similar products in a lower-dimensional space.

**SVD and Product Relationships:**

The SVD technique reduces the number of dimensions in the similarity matrix, helping to identify the most influential latent factors behind product relationships. This allows for:

1. **Identification of Patterns**: SVD identifies key patterns and correlations between products based on user ratings or similarity measures.
2. **Improved Visualization**: The reduced dimensionality aids in better visualizing relationships between products, which would otherwise be hard to discern in the original high-dimensional space.
3. **Enhanced Recommendations**: With fewer but highly influential features, recommendations become more meaningful, as the model captures core product similarities.



Conclusion  
This assignment provided a comprehensive approach to the development of an intelligent recommender system, focusing on data preprocessing, similarity computation, and recommendation generation. The key takeaway is the importance of a structured workflow in cleaning and preparing data to extract meaningful insights. The use of cosine similarity to assess product relationships showed promising results, offering an effective method for content-based recommendations. The project serves as a foundation for future improvements, such as incorporating hybrid recommendation strategies, implementing advanced evaluation metrics (precision, recall, F1-score), and exploring machine learning models for personalized suggestions. Ultimately, this assignment sets the stage for developing more sophisticated and accurate recommendation systems that can handle complex datasets and user interactions.