Analysis of Taste Profile Patterns Using Singular Value Decomposition to Support Restaurant Menu Selection

Natalia Desiany Nursimin - 13523157[1,2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13523157@std.stei.itb.ac.id, [2]nataliadesianyy@gmail.com

***Abstract*— Choosing what to eat in a restaurant often proves to be a challenging task for customers, as the abundance of options creates a "paradox of choice," leading to decision paralysis. This challenge becomes more pronounced when customers visit unfamiliar restaurants, where limited prior knowledge of menu items leads to a harder time in making decisions. This paper presents an applied analytical approach to restaurant menu selection through the analysis of taste profile patterns using Singular Value Decomposition (SVD). Key taste attributes, specifically sweetness, saltiness, sourness, spiciness, umami and richness are quantified and represented in a matrix form, enabling mathematical decomposition and dimensionality reduction. By applying SVD, latent taste dimensions are extracted to reveal hidden relationships between menu items and dominant flavor characteristics. This paper aims to demonstrate how SVD can be effectively applied to solve real world problems, specifically by supporting restaurant menu selection through the matching of customer taste preferences with menu items based on their latent profile representations.**

Keywords—recommendation system; restaurant menu selection; singular value decomposition; taste profile analysis

I. INTRODUCTION

Dining out to eat at a restaurant can often be considered a form of reward to most people. However, it can sometimes become stressful and overwhelming due to the extensive menus, making it hard to decide what to order. As the culinary industry continues to evolve, restaurants have broadened their offerings, frequently presenting diverse cuisines, fusion dishes and lengthy ingredient lists. While this variety aims to cater to a wide range of preferences, it frequently results in a phenomenon known as the "paradox of choice." Psychological research suggests that when faced with an overwhelming number of options, customers can often experience decision paralysis, anxiety or the tendency to default to familiar, safe choices rather than exploring items they might genuinely enjoy.

This challenge is further intensified by the inherent difficulty of predicting sensory experiences based solely on textual descriptions. A menu item labeled “Spicy Tofu Soup,” for example, provides only a generic indication of the dish and fails to communicate critical nuances of its flavor profile, such as the specific balance between acidity and heat or the level of viscosity of the broth. Consequently, customers also face an ordering risk, where the fear of being dissatisfied and spending money on a dish that does not align with their palate discourages experimentation. Without knowing the flavor composition, a diner is forced to guess whether a new dish aligns with their preferences. For example, a customer who enjoys the rich, salty profile of "Sapo Tahu" might not intuitively know that "Sup Tahu Daging Ayam" shares a mathematically similar flavor profile, despite having a different names.

Traditionally, diners rely on waitstaff recommendations to mitigate this risk. However, this approach has limitations. It is inherently inconsistent, prone to personal biases of the server and limited by the human capacity to memorize the complex flavor composition of every dish in a menu book. Furthermore, existing digital solutions often rely on popularity ratings which fail to capture the intrinsic content of the food. For example, a 5-star rating does not explain why a dish is liked, nor does it describe its taste.

To address these limitations, a more objective solution is needed. This paper proposes a mathematical solution to this problem by utilizing Singular Value Decomposition (SVD). SVD is a fundamental technique in linear algebra used for matrix factorization and dimensionality reduction. By representing the menu as a matrix of items versus flavor attributes, SVD allows us to decompose complex taste data into simpler, fundamental components.

The core advantage of applying SVD in this context is its ability to extract latent taste patterns. Rather than treating "Spicy" and "Savory" as isolated variables, SVD identifies hidden structures where these attributes interact, allowing dishes to be grouped based on their flavor DNA. This enables the system to calculate the geometric similarity between a customer’s preference vector and the menu items in a reduced dimensional space, filtering out noise and focusing on the dominant taste characteristics.

This research focuses on the implementation of SVD to analyze a specific dataset of dishes with the aim to construct a “taste matrix” where flavor attributes are quantified numerically. Through this process, the SVD algorithm is applied to reduce the dimensions of the matrix and interpret the resulting latent factors to demonstrate

how mathematical patterns correlate with human sensory perception to support decision making process.

II. Theoretical Background

A. Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a factorization method in linear algebra that decomposes a matrix into three other matrices, enabling data to be represented in terms of its singular values. Formally, SVD factorizes a matrix A of size m x n into three matrices U, Σ, and V^T, represented as the following:

$$A = U\Sigma V^T$$

Where:

- U = is an m x m orthogonal matrix. The columns U are the eigenvectors of AA^T. In the context of this research, the rows of U represent the coordinates of the menu items in the latent taste space.
- Σ = is a m x n diagonal matrix with non-negative real numbers on the diagonal, known as singular values (σ_i), while all other elements are zero. In the context of this research, the magnitude of each singular value indicates the strength or importance of the corresponding latent taste pattern.
- V^T = is a n x n orthogonal matrix (the transpose of V). The columns of V corresponds to the eigenvectors of A^TA. In the context of this research, this matrix represents the coordinates of the flavor attributes in the latent space.

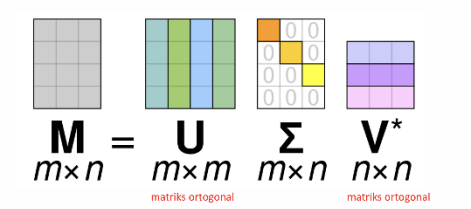


Fig 2.1. Illustration of Singular Value Decomposition

Source:https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-21-Singular-value-decomposition-Bagian1-2025.pdf

B. Quantitative Representation of Taste

To enable mathematical analysis of sensory data, subjective taste characteristics must first be converted into a structured numerical form. In this research, a Vector Space Model is used, where each menu item is represented as a point in a multi-dimensional space. $M = \{m_1, m_2, \dots, m_r\}$ represents the set of menu items. Each item m_i is described by a vector v_i that contains numerical scores corresponding to specific flavor attributes, such as sweetness, saltiness, sourness, spiciness, umami or richness.

$$v_i = [w_i, 1, w_i, 2, \dots, w_i, n]$$

Where:

- w_{ij} = represents the intensity score of the j-th flavor attribute (sweetness, saltiness, sourness, spiciness, umami, richness) in the i-th dish.
- While the full menu book can be expressed as the m x n matrix A.

C. Dimensionality Reduction (Truncated SVD)

While full SVD reconstructs the original matrix perfectly, the goal of pattern recognition is to filter out noise and capture the dominant structures. In this context, it is to filter out minor taste variations and obtain only the dominant features. This can be achieved through the use of truncated SVD. By retaining only the top k singular values and discarding the rest, we can obtain A_k which is the best rank-k approximation of the original matrix A, represented by the following:

$$A_{k=} U_k \Sigma_k V_k^T$$

Where:

- A_k = is the best rank-k approximation of the original data matrix A. In the context of this research, it represents the denoised taste profile matrix, where minor inconsistencies in flavor scoring are filtered out, leaving only the dominant taste structures.
- U_k = is an m x k matrix containing the first k columns of U. In the context of this matrix, it represents the coordinates of the menu items projected into the reduced latent space.
- Σ_k = is a k x k matrix containing the top k singular values. In the context of this matrix, it represents strength of each extracted latent taste pattern.
- V_k^T = is a k x n containing the first k rows of V^T. In the context of this research, this matrix represents the coordinates of the flavor attributes, such as spicy, salty, sweet in the latent space.

D. Latent Semantic Analysis (LSA)

Latent Semantic Analysis (LSA) is a mathematical technique originally developed for Information Retrieval to uncover hidden relationships between items and their attributes. It operates on the assumption that words with similar meanings will occur in similar contexts. In the context of this research, LSA is adapted and remapped to the domain of Computational Gastronomy, represented by the following:

- Documents are replaced by Menu Items
- Terms are replaced by Flavor Attributes, such as Sweetness, Spiciness, Umami

By applying Singular Value Decomposition (SVD) to the Item-Attribute matrix, LSA constructs a Semantic Taste Space. In this reduced space, the system can identify latent flavor concepts, enabling the detection of similarities between dishes based on their overall taste profile.

E. Content-Based Filtering

Content-Based Filtering generates recommendations based on the intrinsic characteristics of the items themselves, rather than relying on historical user ratings and reviews. This approach ensures that each menu item is evaluated objectively according to its flavor profile instead of its popularity among other customers. As a result, lesser-known or newly introduced dishes have the same opportunity to be recommended, provided their flavor representations align with the user's preferences.

The process operates by mapping both the menu items and the user's preference vector into the same k dimensional latent space derived from SVD. Recommendations are then produced by measuring the similarity between the user preference vector and the menu item representations within this latent space. This approach also effectively addresses the Cold-Start Problem, since the recommendation depends solely on the mathematical attributes of the dish rather than historical user data. This allows new menu items to be recommended immediately upon entry into the system without requiring prior user interaction.

F. Cosine Similarity

To help determine the relationship between a user's preference and a menu item, the similarity between their respective vectors in the reduced k-dimensional space needs to be calculated. Cosine Similarity is used as the metric because it measures the cosine of the angle between two vectors, focusing on orientation rather than magnitude. This is represented as the following:

$$similarity\ (a,b) = \frac{a\ .\ b}{||a||.||b||} = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2}\ \sqrt{\sum_{i=1}^{n} b_i^2}}$$

Where:

- 1 indicates identical flavor profiles.
- 0 indicates orthogonal (completely dissimilar) profiles.
- Values close to 1 suggest high similarity

III. METHODOLOGY

Although choosing a meal at a restaurant is a common activity, it often involves a complex decision-making process complicated by the overwhelming number of available options, commonly referred to as the "paradox of choice." Even with detailed menus, customers frequently find it difficult to determine whether a dish will suit their personal preferences, as flavor descriptions are inherently abstract and lack measurable standards. This section of the paper discusses the application of Singular Value Decomposition (SVD) as a systematic mathematical approach to addressing this challenge. By decomposing quantified taste profiles into orthogonal latent components, the proposed method translates subjective culinary preferences into a structured linear algebra framework. The following subsections present the practical implementation of this approach, including taste attribute quantification, matrix factorization and the calculation of geometric similarity to help create a better system to support menu selection.

A. Quantitative Representation of Taste Profiles

The primary obstacle in analyzing taste profiles is the inherent subjectivity of sensory perception. A dish described as spicy by one person might be considered mild by another. To enable mathematical analysis, these qualitative descriptions must be transformed into a structured numerical model. This phase focuses on establishing a uniform evaluation framework. By applying a standardized scoring metric, every menu item regardless of its cuisine type or complexity is measured on an equal footing. This ensures that the resulting comparison between dishes is fair, objective and mathematically consistent.

1. Attribute Taxonomy Definition

Based on culinary science principles, the flavor profile of any dish can be deconstructed into fundamental orthogonal dimensions. This research defines a feature set F consisting of six primary attributes, as follows:

- Sweetness: Intensity of sugar, honey or natural sweetness.
- Saltiness: Intensity of sodium-based components.
- Sourness: Acidity levels derived from vinegar, citrus or fermentation.
- Spiciness: Heat sensation derived from spicy seasonings or chili based ingredients.
- Umami: Savory taste profile derived from glutamates, meat or mushrooms.
- Richness: Perception of viscosity, creaminess fat content or oil.

2. Feature Vectorization

To convert these qualitative attributes into numerical values, a 5-point Likert scale is going to be used to represent the intensity of each attribute. The intensity levels are defined as follows:

- 0: Undetectable (absent)
- 1: Trace amount (subtle)
- 3: Moderate intensity (balanced)
- 5: Dominant characteristic (strong)

For each menu item m_i, a feature vector v_i is constructed as follows:

$$v_i = [w_i, 1, w_i, 2, \dots, w_i, 6]$$

Where w_{ij} represents the intensity score of attributes *j* for dish *i.*

B. Item-Attribute Matrix Construction

The quantified data is aggregated into a unified mathematical structure called the Item-Attribute Matrix(*A).* Where M represents the set of m menu items and *n* represent the number of taste attribute, where *n* = 6. The matrix $A \in R^{mxn}$ is defined as:

$$A = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \cdots & w_{m,n} \end{bmatrix}$$

Each row of this matrix represents the coordinates of a dish in a 6 dimensional flavor space.

C. SVD Decomposition and Dimensionality Reduction

The matrix is decomposed into three matrices, represented as the following:

$$A = U\Sigma\, \mathrm{V}^{\mathrm{T}}$$

Where:

- U (Item-Concept Matrix): an m x m orthogonal matrix relating menu items to latent taste concepts.
- Σ (Singular Value Matrix) : a m x n diagonal matrix containing singular values (σ_i), while all other elements are zero. The magnitude of each singular value indicates how dominant the strength of a specific taste pattern in the dataset.
- V^{T} (Attribute Concept Matrix) = a n x n orthogonal matrix representing relating specific flavor attributes (spicy, sour, salty, umami, sweet, rich) to the latent concepts.

To help filter out the noise (minor taste variations) and fully capture the dominant flavor DNA, the matrix is approximated by retaining only the top *k* singular values. The value of *k* is determined by calculating the Cumulative Energy Ratio, ensuring at least 95% of the variance is preserved which is represented as the following:

$$\frac{\sum_{i=1}^{k} {\sigma_i}^2}{\sum_{i=1}^{n} {\sigma_i}^2} \geq 95$$

This will produce the the rank-*k* approximation $A_{k=}\, U_k \Sigma_k\, {V_k}^{\mathrm{T}}$ which will project the dishes into a lower dimensional latent taste space, where similarity is determined by overall flavor patterns rather than exact attribute values.

D. Recommendation Algorithm

Following the dimensionality reduction process, the system utilizes the extracted latent structures to perform intelligent retrieval tasks. The operational core of the recommendation engine is driven by two sequential operations, designed to align user intent with the established flavor patterns and quantify their resemblance, which are the following:

1. User Preference Projection

User preferences are modeled as a query vector q, which may be obtained either through explicit user input or by selecting a reference dish. Since menu items are represented in the reduced latent space as $U_k\Sigma_k$, the user preference vector are projected into the same space to ensure consistency. The projection is computed as:

$$q_{latent} = q\,.\,V_k\,.\,{\Sigma_k}^{-1}$$

2. Preference Projection

Once both menu items and user preferences are represented in the same latent space, similarity is measured using cosine similarity. This metric evaluates the angular similarity between vectors, focusing on taste orientation rather than absolute intensity. The similarity between a user preference vector and a menu item is defined as:

$$similarity = \cos\theta = \frac{q_{latent}\,.\,m_{latent}}{||q_{latent}||.\,||m_{latent}||}$$

The metric is designed to evaluates the cosine of the angle between two vectors. A value close to 1 indicates that the dish and the user's preference share the same flavor taste orientation. The system then ranks the items in descending order and presents the top *N* recommendations.

IV. IMPLEMENTATION

The program has been designed using Singular Value Decomposition (SVD) and Content-Based Filtering principles to analyze flavor profiles and generate personalized menu recommendations. A Vector Space Model is constructed from sensory attributes divided into six categories: Sweet, Salty, Sour, Spicy, Umami and Richness. The resulting item–attribute matrix is factorized using SVD to uncover latent taste patterns that capture underlying relationships between menu items and flavor characteristics. User preferences are mathematically projected into the same latent space and Cosine Similarity is employed to systematically evaluate the relationship between user preferences and available menu items. This approach enables recommendations to be generated in an objective and mathematically consistent manner.

A. Import Modules

The program utilizes several essential Python libraries to implement the linear algebra and matrix factorization system:

- numpy: implements a high performance multi-dimensional array objects and mathematical operations for matrix multiplication and dot products.
- pandas: manages the structured dataset (DataFrames) for menu attributes, categories and prices.
- scipy.linalg: provides the core linear algebra functions, specifically svd for matrix decomposition and norm for vector normalization.

```
import numpy as np
import pandas as pd
from scipy.linalg import svd, norm
```

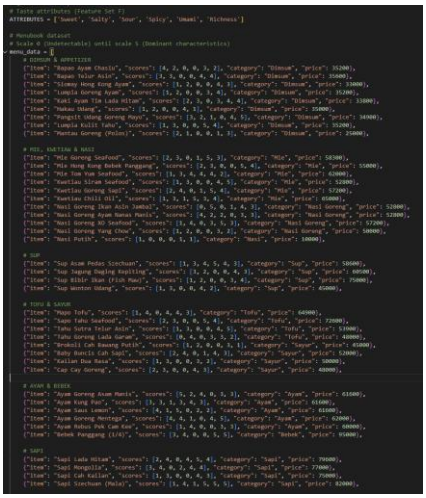
Fig 4.1. Imported Modules

B. System Architecture

The system architecture of the program is divided into four different components, consisting of the following:

1. Core Data Structures: this component consists of the item-attribute matrix representation that forms the mathematical foundation of the sensory analysis.
2. Pattern Extraction Engine (SVD): this component implements the matrix factorization, dimensionality reduction, and latent concept identification using the SVDRecommender class.
3. Recommendation Engine: this component implements the User Projection and Cosine Similarity logic to generate ranked suggestions.
4. User Interface System: this component provides an interactive output mechanism that translates user flavor queries into mathematical vectors and presents results in an easy to read format.

1. Core Data Structures
   - Menu Dataset Initialization
     The system begins by defining the Feature Set (F) containing 6 sensory attributes: Sweet, Salty, Sour, Spicy, Umami and Richness. The dataset employed in this research, specifically regarding item names and prices, is adapted from the official menu of Ta Wan Restaurant. Minor adjustments were applied to specific flavor attributes to enhance the diversity of the taste profiles for experimental purposes. The raw data is structured as a list of dictionaries (menu_data), where each dictionary represents a menu item containing its specific flavor vector (scale 0-5), category and price. The official menu reference can be accessed at: https://www.tawanrestaurant.com/en/our-menus/



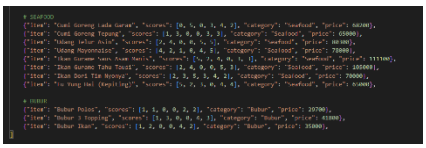


Fig 4.2. Menu Dataset Structure

   - DataFrame Conversion
     The pd.DataFrame function transforms the raw dictionary list into a mathematical matrix format (df_vector) where rows represent items and columns represent flavor attributes.

Fig 4.3. Menu Dataset Structure

2. Pattern Extraction Engine (SVD)

The SVDRecommender class implements the mathematical factorization and dimensionality reduction logic. This component is responsible for transforming the high-dimensional taste data into a condensed Latent Semantic Space.

- fit(self, energy_threshold): This function executes the core SVD algorithm. It decomposes the raw matrix A using scipy.linalg.svd into three components, consisting of U (Left Singular Vectors), Σ (Singular Values) and V^T (Right Singular Vectors). It then calculates the Cumulative Energy of the singular values to automatically determine the optimal rank k that retains at least 95% of the information variance.

- Matrix Truncation and Pre-computation: After determining k, the system truncates the matrices to create the reduced representation. It pre-computes two critical mathematical components for the recommendation phase, which are:
  - Vk: The transposed reduced feature space map used for projecting user queries.
  - Uk_Sigmak: The coordinates of every menu item in the latent space($U_k \cdot \Sigma_k$), enabling easy similarity lookups.

- explain_latent_concepts(self): This function helps map the rows of the V^T matrix to the flavor attributes, allowing the system to identify the dominant latent concepts.

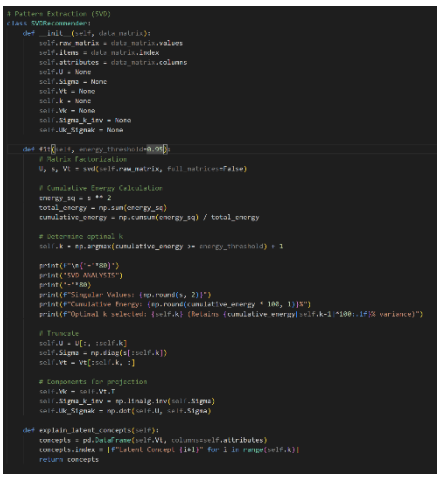


Fig 4.4. Pattern Extraction Engine

3. Recommendation Engine

This component focuses on the implementation of the retrieval and ranking logic to provide menu recommendations based on user preferences.

- recommend_menu(user_input_vector, model): This is the core function that handles the entire recommendation process. It takes the user's taste preferences (for example, [5, 2, 0, 0, 3, 2] representing their desired sweetness, saltiness and other criterias) and runs it through the projection and similarity calculations.

- User Preference Projection
  The system maps the user's raw input vector q into the latent space using the transformation matrix derived from SVD using the following formula: q_latent = np.dot(np.dot(q_raw, model.Vk), model.Sigma_k_inv)

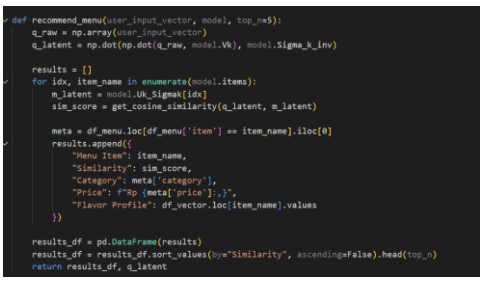


Fig 4.5. User Vector Projection Logic

- get_cosine_similarity(vec1, vec2): This function calculates the alignment between the projected user vector and a menu item vector. It implements the Cosine Similarity formula: $\frac{a \cdot b}{||a||.||b||}$. The higher the score obtained, indicates a more similar match.

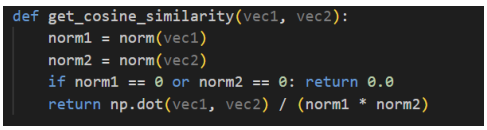


Fig 4.6. Cosine Similarity Calculation

4. User Interface System

The user interface connects what people want to eat with the mathematical recommendation engine. Everything runs through the main() function. The user interface system consists of a few components, such as the following:

- Input Processing (get_user_input) : The system uses a simple command-line interface where users rate how much they want each of the 6 taste attributes on a scale from 0 to 5. The higher the scale number indicates the more dominant they wish to taste that flavor. If someone enters something invalid, like a letter instead of a number or a number outside the 0-5 range, the system catches it and will ask them to try again.

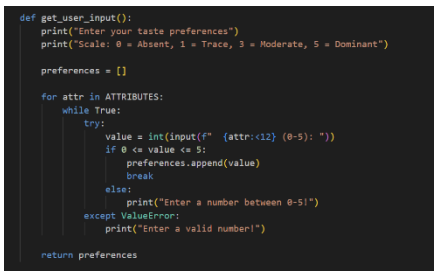


Fig 4.7. User Input

- Preset Scenarios (show_preset_examples): These consists of predefined taste profiles, such as Sweet & Sour Lover, Spicy & Umami Lover, Salty & Rich Lover, Balanced Taste and Mild & Light. This allows users to quickly see how the system responds to distinct taste profiles without manual input.

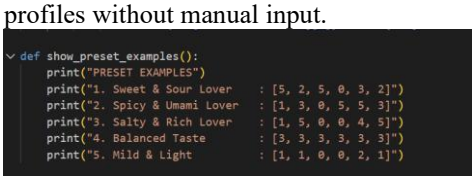


Fig 4.8. Preset Scenarios

- Output Presentation (display_recommendations): The system ranks the items based on their calculated similarity scores in descending order. The system will display the Menu Item, Category, Price, calculated Similarity Score and the detailed Flavor Profile breakdown of the item.

Fig 4.9. Output Presentation

V. TESTING & RESULTS

A. Test Case 1

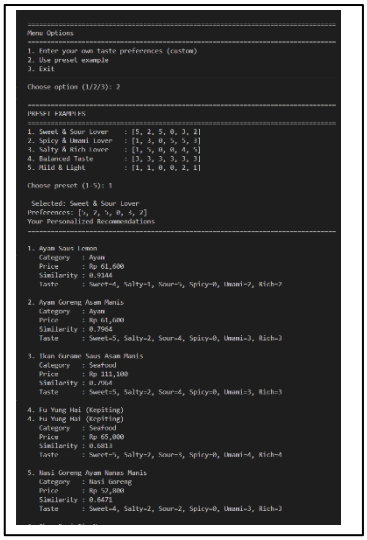


Fig 5.1. Test Result 1

User Input:

- Selected preset: "Sweet & Sour Lover"
- Preference: [5, 2, 5, 0, 3, 2] (Dominant Sweet, Dominant Sour, Moderate Umami)

The result of Test Case 1 shows that the SVD-based recommendation successfully identifies menu items with matching taste profiles. For a user seeking dominant sweet and sour flavors, the system correctly ranked dishes like "Ayam Saus Lemon" and "Ayam Goreng Asam Manis" at the top, with similarity scores ranging from 0.65 to 0.91. All top-5 recommendations share the requested dominant attributes (Sweet & Sour). Notably, "Ayam Saus Lemon" achieved the highest similarity score (0.9144) due to its exact match on the Sour attribute (Sour=5), demonstrating the system's ability to prioritize precise taste profile alignment over semantic name matching.

B. Test Case 2

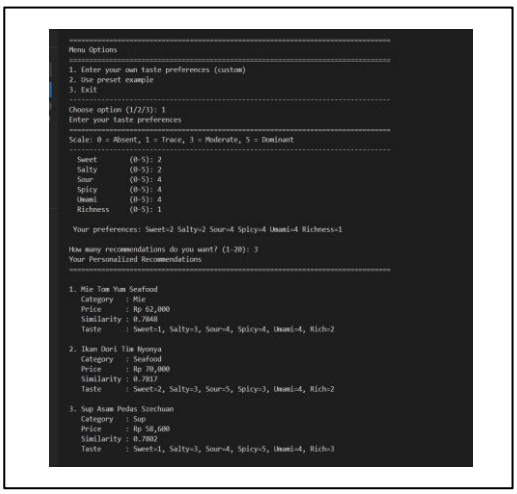


Fig 5.2. Test Result 2

User input:

- Preference: Sweet = 2, Salty = 2, Sour = 4, Spicy = 4, Umami = 4, Richness = 1

The result of test case 2 with a balanced multi-attribute preference profile [2,2,4,4,4,1] successfully identifies menu items satisfying multiple simultaneous taste preferences. The system achieved exceptional performance with "Mie Tom Yum Seafood" ranking first (0.7848), featuring three exact matches (Sour=4, Spicy=4, Umami=4). Tom Yum is a Thai soup renowned for precisely this kind of flavor combination, validating the system's semantic accuracy. The top three recommendations scored within 0.005 similarity points (0.7802-0.7848), indicating equally high relevance. This test case demonstrates that this SVD-based recommendation program excels at multi-attribute optimization, successfully identifying items that satisfy multiple taste preferences simultaneously.

C. Test Case 3

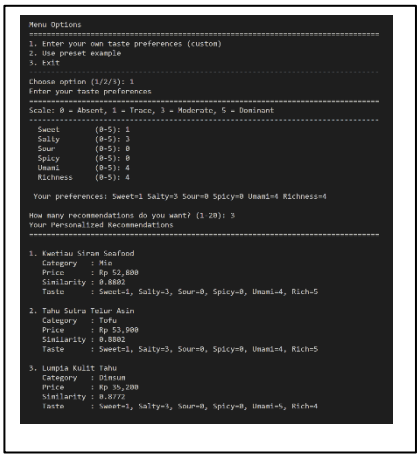


Fig 5.3. Test Result 3

User input:

- Preference: Sweet = 1, Salty = 3, Sour = 0, Spicy = 0, Umami = 4, Richness = 4

The result of Test Case 3 with a custom user input of Umami & Richness preference profile [1,3,0,0,4,4] successfully identifies menu items with matching taste characteristics. The system achieved high similarity scores (0.8772-0.8802). The top recommendations all feature high attributes both in the umami and richness dimensions proving that the system manages to effectively recommend food items aligning with user preferences.

VI. CONCLUSION

This paper has successfully demonstrated that Singular Value Decomposition (SVD), when applied to quantitative taste profiling, offers a robust and objective solution to the "paradox of choice" frequently encountered by people when dining in restaurants. By transforming subjective

sensory attributes, such as sweetness, saltiness, sourness, spiciness, umami, and richness into a structured vector space, the system allows for the mathematical decomposition of flavor complexities. The application of truncated SVD, while retaining 95% of information variance, effectively extracted latent taste dimensions to reveal hidden relationships between menu items, filtering out noise while still managing to preserve critical flavor nuances. The experimental results done have managed to confirm that the usage of SVD in taste profile patterns have accurately aligns user preferences with menu items, successfully distinguishing intricate flavor combinations. Furthermore, the system proves effective in addressing the cold-start problem by relying on intrinsic content rather than historical user data. Ultimately, this research provides evidence that linear algebra techniques can effectively bridge the gap between abstract culinary descriptions and concrete dining decisions, transforming the overwhelming task of menu selection into a personalized, mathematically informed recommendation experience that helps reduces decision fatigue.

VII. Appendix

The GitHub repository for this paper can be accessed at https://github.com/nataliadesiany/makalahalgeo. While the video presentation of this paper is available on YouTube and can be accessed at https://youtu.be/gwbLPHmdaZw?si=HErci3aRpKG3wIdw.

VIII. Acknowledgment

First and foremost, the author wishes to express sincere gratitude to God Almighty for His guidance, blessings and grace, which have made it possible for this paper to be completed on time. The author would also like to express their deepest appreciation to Dr. Ir. Rinaldi Munir, M.T., Ir. Rila Mandala, M.Eng., Ph.D., and Arrival Dwi Sentosa, S.Kom., M.T., for all the guidance and knowledge they have shared throughout the IF2123 Linear Algebra and Geometry course. Furthermore, the author gratefully acknowledges their parents for their constant support, encouragement and motivation during the preparation of this. Lastly, the author thanks all readers of this paper and sincerely hopes that the material presented will prove to be both beneficial and insightful.

References

[1] R. Munir, "Singular Value Decomposition (Bagian 1)," Institut Teknologi Bandung (ITB), 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-21-Singular-value-decomposition-Bagian1-2025.pdf. [Accessed: Dec. 21, 2025].

[2] R. Munir, "Singular Value Decomposition (Bagian 2)," Institut Teknologi Bandung (ITB), 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2025-2026/Algeo-22-Singular-value-decomposition-Bagian2-2025.pdf. [Accessed: Dec. 21, 2025].

[3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009. [Online]. Available: https://www.chrisvolinsky.com/publications/17545-matrix-factorization-techniques-for-recommender-systems. [Accessed: Dec. 22, 2025].

[4] Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, and A.-L. Barabási, "Flavor network and the principles of food pairing," *Sci. Rep.*, vol. 1, Art. no. 196, Dec. 2011. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/22355711/. [Accessed: Dec. 21, 2025].

[5] P. Parvin, F. Rikken, C. Zhang, and S. Boesveldt, "Decoding the Taste Puzzle: Toward a better understanding of taste profiling," *Food Qual. Prefer.*, vol. 129, Art. no. 105474, Aug. 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950329325000497. [Accessed: Dec. 22, 2025].

[6] [7] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system—a case study," in *Proc. ACM WebKDD Workshop*, Boston, MA, USA, 2000. [Online]. Available: https://files.grouplens.org/papers/webKDD00.pdf. [Accessed: Dec. 22, 2025].

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Desember 2025

Natalia Desiany Nursimin
13523157