Food Recommendation System Using Hybrid Filtering

Abdullah Mohammed Hazeq
School of Computing
Asia Pacific University of Technology and Innovation (APU)
57000 Kuala Lumpur, Malaysia
tp070058@mail.apu.edu.my

Ryandra Yudha Bramantyo
School of Computing
Asia Pacific University of Technology and Innovation (APU)
57000 Kuala Lumpur, Malaysia
tp072461@mail.apu.edu.my

Mohammed T.K. Alhabil
School of Computing
Asia Pacific University of Technology and Innovation (APU)
57000 Kuala Lumpur, Malaysia
tp067706@mail.apu.edu.my

Mohammad Shayaan Ismael Dhuny
School of Computing
Asia Pacific University of Technology and Innovation (APU)
57000 Kuala Lumpur, Malaysia
tp071591@mail.apu.edu.my

Dr. Adeline Shena John Chsrisanthum
Senior Lecturer/School of Computing
Asia Pacific University of Technology and Innovation (APU)
57000 Kuala Lumpur, Malaysia
adeline.john@apu.edu.my

Dr. Kamalanathan Shanmugam
Senior Lecturer / School of Technology
Asia Pacific University of Technology & Innovation (APU)
Technology Park Malaysia
57000 Kuala Lumpur, Malaysia.
kamalanathan@apu.edu.my

Thavamalini Sivasamy
Senior Lecturer / School of Technology
Asia Pacific University of Technology & Innovation (APU)
Technology Park Malaysia
57000 Kuala Lumpur, Malaysia.
thavamalini@apu.edu.my

Juhairi Aris Muhamad Shuhili
Asia Pacific University of Technology & Innovation (APU)
Technology Park Malaysia
57000 Kuala Lumpur, Malaysia.
juhairi.shuhili@apu.edu.my

***Abstract*— Recommendation systems are widely employed across various domains. Machine learning is currently a prominent trend in the field of data analysis. This paper incorporates recommendation systems into machine learning, specifically utilising the AI techniques of Collaborative Filtering and Content-based Filtering (Hybrid Filtering). The study paper also includes a demonstration of the findings and alternatives to improve the accuracy of the said findings. The journal references a handful of research papers written by proficient authors to enhance and strengthen the information and claims created in the journal.**

Keywords—Hybrid Filtering, Recommendation System, Content-Based Filtering, Food, TF-IDF, Cosine Similarity

I. INTRODUCTION

Currently, an immense quantity of information is easily accessible to us. The tremendous amount of information available might have negative consequences, such as information overload and heightened decision-making challenges caused by the extensive complexity and many possibilities to consider. To mitigate this problem, a recommendation or recommender system was developed. Nevertheless, the recommendation system encounters a myriad of new challenges. An inherent challenge faced by this system is the cold start problem. The complicated start problem arises when there is insufficient data on a user or object to generate a precise recommendation (Yuan & Hernandez, 2023). Several methodologies can address this issue: content-based, collaborative filtering (*COLLABORATIVE FILTERING*, 2022), hybrid filtering, TF-IDF and cosine similarity. This research evaluates various strategies to address the cold start problem.

II. LITERATURE REVIEW

A. Similar Projects

Scholars have explored various topics related to recommender systems, with a specific emphasis on suggestions for food. For example, Petrusel and Limboi used sentiment analysis with recommendation engines to suggest eateries based on favourable and unfavourable customer reviews. Sawant and Pai used content-based and collaborative characteristics from the Yelp dataset to create a Yelp meal recommendation system.

Yera and colleagues refined a food recommendation system that considers user preferences and nutritional data. Maia and Ferreira investigated a context-aware meal recommendation system that considers food items' location and cooperative filtering. In their investigation of user-reviewed recommender systems, Chen et al. emphasised the value of user-generated material in improving suggestions (Maiyaporn Phanich et al., 2010).

Several initiatives and studies have focused on diabetic meal recommendation systems. These programs encourage healthy eating and nutritional management. With a hierarchical clustering technique, (Li et al., 2016) created an ontology-based food categorising system. Their technique featured three nutritional levels (low, medium, and high) but was not optimised for diabetic diet management and had manual setup constraints. A customised diabetic food recommendation system by Lee et al. (2018) used ontology and fuzzy logic to prioritise supper calories over food category substitutions (Singh & Dwivedi, 2023).

B. Approach

The authors adopted a hybrid strategy in their study, skillfully fusing collaborative filtering techniques with content-based filtering to provide food suggestions. User-specific characteristics, including meal name, ID, cuisine, and diet type, were targeted using content-based filtering. Textual data related to food products, especially cosine similarity, was preprocessed and analysed using string-matching methods (Maiyaporn Phanich et al., 2010).

The collaborative filtering system considered meal names and user ratings to make predictions. Cosine similarity was used to create a matrix that captured user-item interactions and measured how similar the user and food IDs were. This method also considered linguistic data associated with food products by turning them into feature vectors (Maiyaporn Phanich et al., 2010).

The Food Recommendation System (FRS) for people with diabetes by Maiyaporn Phanich et al. uses SOM and K-means clustering. Using a Thai food nutritive values dataset, data is prepared and categorised by features and diabetic appropriateness. Feature extraction ranks eight essential nutrients for people with diabetes. The clustering study uses SOM and K-means clustering for unsupervised learning and visualisation. Based on nutritional combinations and features, Recommended Foods (RF) categories are created from results and clusters. The Food Recommendation System uses Distance Matrices to make cluster-based suggestions based on proximity and nutritional similarity (Singh & Dwivedi, 2023).

C. Conclusion

The study presented a complete meal recommender system that elegantly combines collaborative and content-based filtering techniques. String-matching ideas were added to content-based filtering, which improved the system's ability to identify textual similarities and produce more individualised and accurate meal suggestions based on user preferences. Although the model accomplished its goals, the authors are aware of its shortcomings, including its reliance on publicly available statistics and the possibility that it overlooked elements like temporal dynamics or social influence. Future improvements may include adding other variables, such as nutritional data, ingredient preferences, dietary restrictions, or user demographics, to deliver even more individualised and precisely customised suggestions (Maiyaporn Phanich et al., 2010).

The Food Recommendation System (FRS) for diabetics, which uses clustering analysis with Self-Organizing Map (SOM) and K-means clustering, seems promising for helping people make healthier food choices. Clustering foods with nutrients and features meet diabetics' dietary demands. Its capacity to offer same-category alternatives makes it useful for various diets. Enhancing the recommendation algorithm, adding user input, and increasing the dataset might optimise the system. Collaboration with healthcare experts is essential for diabetic diet care validation and accuracy improvement. The Food Recommendation System is a vital and creative resource that improves diabetes patients' nutrition treatment and health (Singh & Dwivedi, 2023).

III. MATERIALS AND METHODS

A. Abbreviations and Acronyms

- Content-based Filtering
- Collaborative Filtering
- Hybrid Filtering

Combined, content-based and collaborative filtering result in Hybrid Filtering. This ML algorithm filters out heaps of items to a specific set of items based on the user’s preference or liking. The recommendation system makes recommendations based on the user’s desire for what food item they want to eat at that moment. Then, the items are ranked based on reviews of another user, which is a part of collaborative filtering. The source code was developed from scratch while referring to a study posted by ‘GRACE HEPHZIBAH M’ on Kaggle.com, including the datasets used for this paper.

B. Software Requirements

- Python
- Python IDE

IDEs tend to make it easier and produce the least number of errors compared to GoogleColab, which is why it is recommended to use a Python IDE.

C. Materials

- Python Libraries: The script uses several libraries, including numpy, pandas, sklearn, nltk, and. These libraries provide various functions and methods that are used throughout the script. Nltk is for tokenisation, sklearn is for the machine learning library, pandas are for data manipulation, and numpy is for numerical computing.
- Dataset: The script uses a dataset of food items and their descriptions named “Food.csv”. The dataset is loaded from a specified file path into a pandas DataFrame. The dataset should have specific columns such as “Food_ID”, food name as “Name”, cuisine type as “C_Type”, and the description as

"Describe", and "Tags", as expected by the script. This dataset is later merged with another dataset named "Ratings.csv," which will hold ratings of users and rank the recommendations for the user to help assist the user in picking a dish.

- User Input: The script takes user input, the user enters a food item, and the script recommends dishes based on this input.

D. Methods

The method used in this Python script is Hybrid Filtering for recommendation systems. Here is a breakdown of the process:

- Data Preprocessing: The data is pre-processed by splitting and joining various columns. This is done to prepare the data for the recommendation system. The two datasets being merged are "Food.csv" and "Ratings.csv". The program then takes this newly merged data frame (df), tokenises it and makes all the tags lowercase. Finally, all of this is then put into a new data frame.

- Feature Extraction: The TF-IFD Vectorizer (Ma et al., 2022) from Sklearn converts the tags into numerical vectors. This process is also known as vectorisation (Rafael Martins D'Addio et al., 2017). Be aware of the different meanings of the homophones "affect" and "effect", "complement" and The stop_words="English" parameter removes common English words (like 'the', 'is', 'in') that do not contain significant meaning and are often removed from texts (Chen et al., 2017).

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Figure 1: TF-IDF formula (TF-IDF, 2024)

- Cosine Similarity: The cosine similarity between the vectors is calculated. This is used to find similar dishes based on the user's input. Cosine similarity is a measure between two non-zero vectors of an inner product space that measures the cosine of the angle between them (Miesle, 2023).

$$\cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figure 2: Cosine Similarity formula (Varun, 2020)

- Tokenization: The word_tokenize (Pai, 2020) from nltk reduces sentences into a set of words. For example, in the sentence "Rob was fishing", the sentence gets tokenised, so the sentence would look something like this ["Rob", "was", "fishing"]. This process is known as tokenisation (Admin, 2023). As the dataset chosen is full of sentences, with each word being an ingredient, tokenisation is preferred to stemming (Perry, 2021).

- Recommender Function**:** This function (recommender) takes a food item as input (food) and returns a list of dishes. The function uses the cosine similarity to find the most similar dishes while also using the mean of the average ratings of a dish to propose what dish is more preferred over the other dishes. The list is given in descending order in terms of recommended dishes.

$$\text{average\_ratings}(f) = \frac{1}{|R(f)|}\sum_{r\,\in\,R(f)} r$$

Figure 3: Average Ratings formula

The method used in this script is a common approach in building recommendation systems, especially when dealing with text data. It is worth noting that the method's effectiveness can depend on the quality and quantity of the data used.

IV. Algorithm Implementation

A. Purpose

The purpose of this Python script and the method used is to build a food recommendation system.
When a user inputs a food item or type they like, the system recommends similar dishes based on the input. This is particularly useful in scenarios such as a food delivery app or a recipe website, where users might look for new dishes similar to their favourites. The method used, which involves data preprocessing, feature extraction, cosine similarity calculation, and tokenisation, allows the system to understand the characteristics of different dishes and find similarities between them. This way, when a user inputs a food item, the system can find dishes with similar traits and recommend them to the user.

In summary, this script aims to provide personalised food recommendations to users based on their preferences, enhancing their culinary experiences and making it easier to discover new dishes they might enjoy.

B. Parameters

The parameters used in the code are mainly the columns of the dataset, such as "Food_ID", "Name", "Tags", and "ratings" majorly. These are used to process the data and make recommendations. The food parameter in the recommender function is the food item input by the user. Functions are used to preprocess the data frame, which was formed by merging the two datasets and inserting them into a new data frame for the primary recommendation function.

V. Results and Discussion

The results presented in this study will result from all the methods involved, including Tokenization, Pre-processing of data, Feature extraction, cosine similarity and the Recommendation function. A set of assumptions must be set to determine what environment the recommendation system will provide desired results. The following assumptions are being made before the actual execution of the program.

- The user does not precisely know what food he wishes to eat.
- The user has a specific type of food in mind but does not precisely know what dish he wishes to consume.
- The user would like to know what dish is preferred over the other dish by other users.

A. *Discuss the implementation*

1) *Setup on Integrated Development Environment*

To start up the implementation of the program, we will need an IDE. The study was conducted on a Python Integrated Development Environment (IDE) named PyCharm, as it was the most convenient platform on which to conduct our study. PyCharm provided a more effortless and smoother experience in installing packages. The packages needed to run the program successfully are Numpy, Pandas, Sklearn, and NLTK, discussed in detail above. The next step would be to download the CSV files, which are both provided on Kaggle.com. The files were renamed "Food.csv" and "Ratings.csv" for convenience. The Food.csv file contains the Food_ID, Name, Veg_Non and Describe (Ingredients), while Ratings.csv contains Food_ID, User_ID and Rating. The Food.csv file alone would provide a sufficient Content-Based recommendation system, and adding the Ratings.csv file would create a Collaborative Recommendation system. As mentioned before, the Content-Based and Collaborative filtering would result in Hybrid Filtering, the machine learning algorithm used for this study.

As the setup phase is completed, now is to talk about how the program runs. The program first merges the CSV files into a data frame that other functions will use. After the merging, the data frame is pre-processed by tokenising words in the columns, thereby creating a new column named "Tags". The column "Tags" is further processed to turn them into lowercase, which will avoid any errors or issues during the running of the program and after all the processing is done, the "Tags" column is then inserted into a new data frame. This newly formed data frame is later used in the calculate similarity function, which converts the data frame into a TF-IDF matrix using TF-IDF Vectorizing. The calculate similarity function also calculates the cosine similarity between TF-IDF Vectors and returns the resulting value.

Moving on to the main central part of the program, the Recommender Function. This function takes the user input, the new data frame, the similarity matrix, and the original data frame. Find foods containing the input keyword and check the cosine similarity distance between the input food and other foods to provide suggestions. The list is provided in terms of each food's mean ratings. The higher the mean rating of a food in a list of recommendations, the higher it is displayed on the list.

2) *Challenges*

The main challenge is finding the perfect max_feature of TF-IDF to ensure the recommendations are accurate. Moreover, during the several trials, the accuracy increased from 50% to somewhere around 70% or more. Multiple changes had to be made. Previously, stemming was used in place of tokenization, which made the data hard to understand and retrieve for the recommendation system.

```
Stemmed Tags:
summer squash salad healthi food veg whitebalsamicvinegar lemonju lemonrind redchi
summersquash(zucchini) seasalt blackpepp basilleav
summer squash salad healthi food veg whitebalsamicvinegar lemonju lemonrind redchi
summersquash(zucchini) seasalt blackpepp basilleav
chicken minc salad healthi food non-veg oliveoil chickenminc garlic(minced) onion salt
sweetchillisauc peanutbutt ginger soysauc freshcilantro redpepperflakes(crushed) tart
chicken minc salad healthi food non-veg oliveoil chickenminc garlic(minced) onion salt
sweetchillisauc peanutbutt ginger soysauc freshcilantro redpepperflakes(crushed) tart
chicken minc salad healthi food non-veg oliveoil chickenminc garlic(minced) onion salt
sweetchillisauc peanutbutt ginger soysauc freshcilantro redpepperflakes(crushed) tart
sweet chilli almond snack veg almondswhol eggwhit curryleav salt sugar(finegrain) red
sweet chilli almond snack veg almondswhol eggwhit curryleav salt sugar(finegrain) red
tricolour salad healthi food veg vinegar honey/sugar soysauc salt garliccloves(minced)
carrot(peeled) cucumb mintleav toastedpeanut
tricolour salad healthi food veg vinegar honey/sugar soysauc salt garliccloves(minced)
carrot(peeled) cucumb mintleav toastedpeanut
christma cake dessert veg christmasdryfruits(pre-soaked) orangezest lemonzest jagge
egg
christma cake dessert veg christmasdryfruits(pre-soaked) orangezest lemonzest jagge
egg
christma cake dessert veg christmasdryfruits(pre-soaked) orangezest lemonzest jagge
egg
christma cake dessert veg christmasdryfruits(pre-soaked) orangezest lemonzest jagge
```

Figure 4: Stemming used for Data frame

Implementing linear kernel and several other techniques to increase the accuracy proved ineffective as they would introduce more issues into the program.

B. *Results*

The results of this study will be based on part-to-part success, where we will check out if the code works at all points of the code as planned. We will start by merging both CSV files into 1 data frame. To check if the merging succeeded, we would use the code df.info ().

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 511 entries, 0 to 510
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Food_ID   511 non-null    int64
 1   Name      511 non-null    object
 2   C_Type    511 non-null    object
 3   Veg_Non   511 non-null    object
 4   Describe  511 non-null    object
 5   User_ID   511 non-null    float64
 6   Rating    511 non-null    float64
 7   Tags      511 non-null    object
dtypes: float64(2), int64(1), object(5)
memory usage: 32.1+ KB
```

Figure 5: Proof of Data frame merging using df.info()

This result shows that the merging succeeded, as columns from both CSV files can be seen in 1 data frame. This allows the program to move to the data processing stage, where the recommender function will tokenize the data.

```
Tokenized Tags:
summer squash salad Healthy Food veg white balsamic vinegar , lemon juice , lemon rind , red
summer squash salad Healthy Food veg white balsamic vinegar , lemon juice , lemon rind , red
chicken minced salad Healthy Food non-veg olive oil , chicken mince , garlic ( minced ) , on
chicken minced salad Healthy Food non-veg olive oil , chicken mince , garlic ( minced ) , on
chicken minced salad Healthy Food non-veg olive oil , chicken mince , garlic ( minced ) , on
sweet chilli almonds Snack veg almonds whole , egg white , curry leaves , salt , sugar ( fin
sweet chilli almonds Snack veg almonds whole , egg white , curry leaves , salt , sugar ( fin
tricolour salad Healthy Food veg vinegar , honey/sugar , soy sauce , salt , garlic cloves (
tricolour salad Healthy Food veg vinegar , honey/sugar , soy sauce , salt , garlic cloves (
christmas cake Dessert veg christmas dry fruits ( pre-soaked ) , orange zest , lemon zest ,
christmas cake Dessert veg christmas dry fruits ( pre-soaked ) , orange zest , lemon zest ,
christmas cake Dessert veg christmas dry fruits ( pre-soaked ) , orange zest , lemon zest ,
christmas cake Dessert veg christmas dry fruits ( pre-soaked ) , orange zest , lemon zest ,
christmas cake Dessert veg christmas dry fruits ( pre-soaked ) , orange zest , lemon zest ,
christmas cake Dessert veg christmas dry fruits ( pre-soaked ) , orange zest , lemon zest ,
japanese curry arancini with barley salsa Japanese veg japanese curry , sticky rice , cheese
japanese curry arancini with barley salsa Japanese veg japanese curry , sticky rice , cheese
japanese curry arancini with barley salsa Japanese veg japanese curry , sticky rice , cheese
japanese curry arancini with barley salsa Japanese veg japanese curry , sticky rice , cheese
chocolate nero cookies Dessert veg almonds , eggs , granulated sugar , bittersweet chocolate
```

Figure 6: Tokenizing for Data frame

The above figure indicates that the tokenisation process was a success, and the data frame "Tags" is ready to be processed by the recommender function to finally provide recommendations to the user.

```
User: veg
Chatbot: Here is a list of dishes with veg, sorted by average rating:
1. summer squash salad
2. amaranthus granola with lemon yogurt, berries and marigold
3. chicken minced salad
4. green cucumber shots
5. vegetable som tam salad
6. shepherds salad (tamatar-kheera salaad)
7. veg summer rolls
8. shrimp & cilantro ceviche
9. beetroot and green apple soup
```

Figure 7: Recommended dishes based on what the user requests

The above figure shows the user asking for vegan (veg) food recommendations. As stated in our assumptions, the user does not know what he wants to eat but knows what type of food he wants. As the results indicate, 22.22% of results are irrelevant to the type of food asked. After leaving a margin for error, the program's success rate, as stated before, is around 75%, which is much better than the past trials. The retrieval of these recommendations is solely through cosine similarity.

```
Cosine Similarity Values:
Food_ID: 1, Cosine Similarity: 1.0000000000000002
Food_ID: 221, Cosine Similarity: 1.0000000000000002
Food_ID: 2, Cosine Similarity: 0.17024768583835292
Food_ID: 164, Cosine Similarity: 0.17024768583835292
Food_ID: 28, Cosine Similarity: 0.17024768583835292
Food_ID: 70, Cosine Similarity: 0.00341939360857988
Food_ID: 223, Cosine Similarity: 0.00341939360857988
Food_ID: 144, Cosine Similarity: 0.12194919912100385
Food_ID: 79, Cosine Similarity: 0.12194919912100385
```

Figure 8: Cosine Similarity for Figure 7

These food IDs represent the food items in *Figure 7*. Each has a cosine similarity in contrast to the user input to other food dishes.

CONCLUSION

In conclusion, Recommendation systems are vital tools for making informed decisions. Recommendation systems have difficulties addressing the cold start problem, which arises when insufficient data is available on new users to provide accurate item recommendations. By adopting collaborative filtering, users can gradually enhance the precision of their suggestions by considering their encounters with food and their resemblance to other users.

ACKNOWLEDGEMENT

The writers of this journal would like to show gratitude towards the user 'GRACE HEPHZIBAH M' on Kaggle.com as they took inspiration from analysing the user's source code, which was posted online. Another massive thanks towards the authors who have poured their passion and knowledge into the referenced research journals, allowing the writers to strengthen their understanding of recommendation systems. Also, I want to express my profound appreciation to all the members who have contributed their best efforts to this research paper. Furthermore, the writers would like to thank the Asia Pacific University of Technology & Innovation (APU) for the encouragement and platform to express knowledge through this journal.

REFERENCES

Yuan, H., & Hernandez, A. A. (2023, December 3). *User Cold Start Problem in Recommendation Systems: A Systematic Review*. ResearchGate; IEEE (Institute of Electrical and Electronics Engineers). https://www.researchgate.net/publication/376140792_User_Cold_Start_Problem_in_Recommendation_Systems_A_Systematic_Review

Food recommendation system using clustering analysis for diabetic patients. (2010, April 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/5480416/

Food recommendation systems based on content-based and collaborative filtering techniques. (2023, July 6). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/10307080/

Perry, T. (2021, February). *What is Tokenization in Natural Language Processing (NLP)? - Machine Learning Plus*. Machine Learning Plus. https://www.machinelearningplus.com/nlp/what-is-tokenization-in-natural-language-processing/

Pai, A. (2020, May 25). *What is Tokenization in NLP? Here is All You Need To Know*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/

Admin. (2023, May 3). *What is Tokenization in NLP (with Python Examples) | PythonProg*. PythonProg. https://www.pythonprog.com/tokenization/

Attaluri, S., Batcha, N., & Mafas, R. (2020). Crop Plantation Recommendation using Feature Extraction and Machine Learning Techniques. *Journal of Applied Technology and Innovation*, *4*(4), 1. https://dif7uuh3zqcps.cloudfront.net/wp-content/uploads/sites/11/2020/11/02171317/Crop-Plantation-Recommendation-using-Feature-Extraction-and-Machine-Learning-Techniques.pdf

Miesle, P. (2023, September 29). *What is Cosine Similarity? A Comprehensive Guide*. DataStax; DataStax. https://www.datastax.com/guides/what-is-cosine-similarity

Rafael Martins D'Addio, Marcos Aurélio Domingues, & Manzato, M. G. (2017). Exploiting feature extraction techniques on users' reviews for movies recommendation. *Journal of the Brazilian Computer Society*, *23*(1). https://doi.org/10.1186/s13173-017-0057-8

Chen, H.-W., Wu, Y.-L., Hor, M.-K., & Tang, C.-Y. (2017). *Fully content-based movie recommender system with feature extraction using neural network*. https://doi.org/10.1109/icmlc.2017.8108968

Ma, M., Wang, G., & Fan, T. (2022). Improved DeepFM Recommendation Algorithm Incorporating Deep Feature Extraction. *Applied Sciences*, *12*(23), 11992–11992. https://doi.org/10.3390/app122311992

Collaborative Filtering. (2022). Google for Developers. https://developers.google.com/machine-learning/recommendation/collaborative/basics

Chhipa, S., Berwal, V., Hirapure, T., & Banerjee, S. (2022). Recipe Recommendation System Using TF-IDF. *ITM Web of Conferences*, *44*, 02006. https://doi.org/10.1051/itmconf/20224402006

gracehephzibahm. (2023, January 7). *Food Recommendation System | Easy & Comprehensive*. Kaggle.com; Kaggle. https://www.kaggle.com/code/gracehephzibahm/food-recommendation-system-easy-comprehensive

Lim, J. J. E., Chan, V. J., Chong, K. L., Salam, Z. A. A., & Lim, D. L. Z. Y. (2024). Product recommendation system using SVD and machine learning techniques[1]. Journal of Applied Technology and Innovation, 8(1), 21-24. https://jati.sites.apiit.edu.my/files/2024/01/JATI_Vol8_Issue1_Paper2_2024.pdf[2]