

# **Time Aware Food Recommender System based on Deep Learning and Graph Clustering**

A SEMINAR REPORT

submitted

*By*

NOEL JOHN ROBERT (B20CS1147, MBT20CS090)

to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the award of the Degree

of

**BACHELOR OF TECHNOLOGY**

IN

*COMPUTER SCIENCE AND ENGINEERING*

**December 2023**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

MAR BASELIOS COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous)

Bethany Hills, Nalanchira

Thiruvananthapuram 15

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**MAR BASELIOS COLLEGE OF ENGINEERING & TECHNOLOGY**

(Autonomous)

Nalanchira, Thiruvananthapuram.



**CERTIFICATE**

*This is to certify that the seminar report entitled **Time Aware Food Recommender System based on Deep Learning and Graph Clustering** submitted by **Noel John Robert, B20CS1147, MBT20CS090** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering and Technology is a bonafide record of the seminar work carried out by him under my guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose*

Seminar Coordinator  
Ms. Asha S

Internal Supervisor  
Dr. Tessy Mathew

Head of the Department  
Dr. Jisha John

# Acknowledgement

First and foremost, I thank God Almighty for their divine grace and blessings in making all these possible. It is to render my heartfelt thanks and gratitude to my principal, Dr. Abraham T. Mathew, for providing the opportunity to do this seminar during the 7th semester of my B. Tech degree course.

Next, I would like to express my sincere gratitude to my seminar coordinator **Ms. Asha S** of the Department of Computer Science and Engineering, for her support and encouragement. I also thank my guide **Dr. Tessy Mathew** of the Department of Computer Science, for her valuable guidance and support. I also thank all the staff members of the Department of Computer Science and Engineering for providing their assistance and support.

Last but not least, I thank all my friends and my family for their valuable feedback as well as their help and encouragement.

# ABSTRACT

Recommender systems are important tools that deliver personalized content, and help recommend relevant products to customers in various domains including e-commerce, social media, and entertainment. Food recommendation systems provide an effective tool that can help users adjust their eating habits and provide a healthier diet.

Previous food recommender systems have shown to be effective in learning a user's pattern by mapping previous interactions with food, but they still suffer some drawbacks, including: ingredients of food, time factor, cold start users and cold start foods, user's community.

The system plans to overcome all these using a method called as "Time-aware food recommender-system based on Deep Learning and Graph Clustering [TDLGC]" [1]. It involves two phases:

(i) recommendation based on users, and (ii) recommendation based on contents of food. First phase considers a similarity-matrix of the user to predict ratings and the second phase utilizes a deep-learning based clustering algorithm to group the foods into several clusters and then rating of unseen foods is predicted.

This model has several novelties, such as: (i) ingredients-aware food recommender system, (ii) time-aware food recommender system, (iii) trust-aware food recommender system, (iv) community-aware food recommender system. Graph clustering is used to group nodes in a graph into clusters based on their similarity and connectivity. Edge weights are calculated using the similarities between user ratings. Deep learning uses a neural network in-order to learn the features of the users and food items. The time factor in this system is used to take into account the changes in food preferences over time.

# Contents

|   |           |
|---|-----------|
| <b>List of Figures</b>                        | <b>iv</b> |
| <b>Nomenclature</b>                           | <b>v</b>  |
| <b>1 Introduction</b>                         | <b>1</b>  |
| <b>2 Methodology</b>                          | <b>2</b>  |
| 2.1 User Based Rating Prediction . . . . .    | 2         |
| 2.1.1 User Similarity Calculation . . . . .   | 2         |
| 2.1.2 Generate Trust Network . . . . .        | 3         |
| 2.1.3 Graph Representation of Users . . . . . | 4         |
| 2.1.4 User Clustering . . . . .               | 4         |
| 2.1.5 User-based Rating Prediction . . . . .  | 4         |
| 2.2 Food Based Rating Prediction . . . . .    | 6         |
| 2.2.1 Food Deep Embedding . . . . .           | 6         |
| 2.2.2 Food Similarity Calculation . . . . .   | 6         |
| 2.2.3 Food Clustering . . . . .               | 6         |
| 2.2.4 Food-based Rating Prediction . . . . .  | 8         |
| 2.3 Top-N Recommendation . . . . .            | 8         |
| <b>3 Conclusion</b>                           | <b>10</b> |
| <b>References</b>                             | <b>11</b> |

# List of Figures

|     |  |   |
|-----|--|---|
| 2.1 | Conceptual framework of the model . . . . .                                      | 3 |
| 2.2 | User-based rating prediction model for sample dataset with nine users . . . . .  | 5 |
| 2.3 | Overview of BERT-based food embedding method . . . . .                           | 7 |
| 2.4 | Food-based rating prediction model for sample dataset with seven users . . . . . | 9 |

## Nomenclature

1. TDLGC - Time Aware Food Recommender System based on Deep Learning and Graph Clustering
2. BERT - Bidirectional Encoder Representation from Transformer
3. NLP - Natural Language Processing

# Chapter 1

## Introduction

The advent of the internet and digital technologies has revolutionized the way we access information and make decisions. One area where this impact is significantly felt is in the domain of food choices. With a plethora of options available, making a healthy and personalized food choice can be overwhelming. This is where food recommender systems come into play. They aim to provide personalized food recommendations that align with the user's preferences, dietary needs, and health goals.

However, existing food recommender systems have several limitations. They often ignore crucial factors such as the ingredients of the food, the time factor (evolution and change in user tastes and diets over time), cold start users and foods (which have insufficient ratings or interactions), and user communities (which can improve the quality and diversity of recommendations). These limitations can lead to recommendations that are not truly personalized or beneficial to the user.

Food recommender systems are categorized into three types:

- Collaborative Filtering - where items are suggested based on the preferences and behavior of similar users. Yummly is an example of this, where recipes liked by users with similar tastes are recommended.
- Content-based Filtering- where items are suggested after their attributes are analyzed and matched with the user's past interaction. Foodpairing is an example, where recommendation is done based on the flavor profile of ingredients.
- Hybrid model - which combines the above two methods for providing a more accurate and diverse recommendations. Examples are OpenTable and HelloFresh.

TDLGC is a hybrid food recommender system that combines collaborative filtering and content-based filtering approaches to address the limitations of existing systems.



# Chapter 2

## Methodology

In TDLGC, the recommendation process is split into two phases:

- First step is User-based rating prediction, where user's community and similarity matrices are used in order to predict user-based ratings[1].
- Second step is Food-based rating prediction, where food items are grouped into several clusters using a deep-learning based clustering algorithm[1, 2] from which ratings on unseen foods are predicted.

Following these two steps, the Top-N foods are recommended[1] to the user.

Fig 2.1[1, 2] shows the model's conceptual framework, showcasing two distinct phases: User-based rating prediction and Food-based rating prediction[1]. In the first phase, the model leverages user ratings along with the follower-following network in order to generate user-user similarity matrix[1, 2] and a trust network. Then, a graph clustering algorithm[3] is used to group the users into different communities. Finally, it predicts the ratings of unseen foods for each user based on their cluster, similarity, and trust information. In the second phase, the model uses deep learning techniques to embed food ingredients into a latent space. Then, it calculates the similarity between different foods based on their embeddings. Finally, ratings on unseen foods for each user is predicted based on food similarity and historical ratings. After the two phases, the model combines the user-based and food-based rating predictions to generate a Top-N food recommendation list for each user.

### 2.1 User Based Rating Prediction

#### 2.1.1 User Similarity Calculation

Similarity measure between users is calculated using cosine similarity formula, which also takes into account ratings and time stamps of the user[1].

$$sim(u_i, u_j) = \frac{a}{bc}, \quad (1)$$

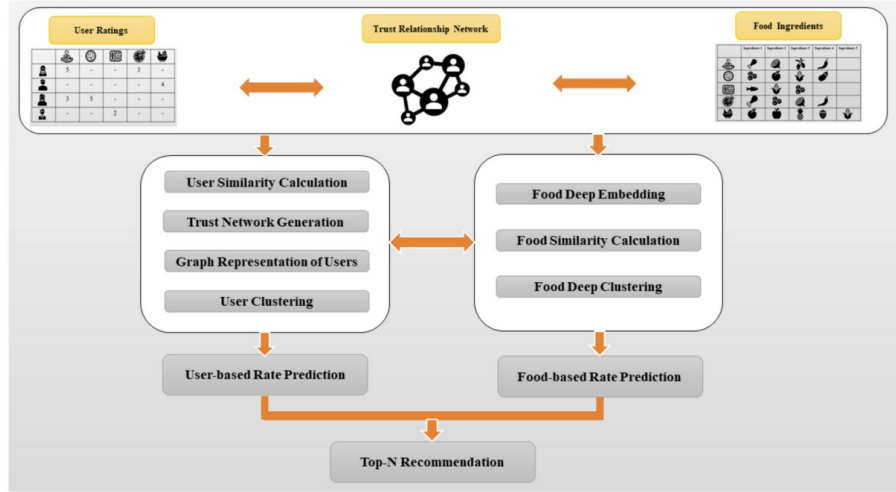


Figure 2.1: Conceptual framework of the model

where values of  $a$ ,  $b$ , and  $c$  are found using the given formulae[1, 2]:

$$\begin{cases} a = \sum_{k \in A_{u_i, u_j}} ((r_k(u_i) - \bar{r}(u_i)) \cdot (r_k(u_j) - \bar{r}(u_j)) TW_{(u_i, u_j, k)}), \\ b = \sqrt{\sum_{k \in A_{u_i, u_j}} ((r_k(u_i) - \bar{r}(u_i))^2 \cdot TW_{(u_i, u_j, k)})}, \\ c = \sqrt{\sum_{k \in A_{u_i, u_j}} ((r_k(u_j) - \bar{r}(u_j))^2 \cdot TW_{(u_i, u_j, k)})}. \end{cases}$$

Here,  $TW_{(u_i, u_j, k)}$  represents the cumulative weights assigned to ratings of users  $u_i$  and  $u_j$  for food  $f_k$  after factoring in time stamps of the ratings[1]. This weight is calculated as given in the below formula[1]:

$$TW_{(u_i, u_j, k)} = e^{-\lambda(TP - t(u_i, k))} + e^{-\lambda(TP - t(u_j, k))} \quad (2)$$

$\lambda$  is a parameter which is controlled by the user in order to adjust the influence of time factor in calculating accumulative weight.  $TP$  indicates maximum Time Period.

### 2.1.2 Generate Trust Network

Trust network is a graph that represents the trust relationship among users. Some studies have shown that users who trust each other, will show profiles with similar ratings[4]. The assumption is made that if user  $u_i$  follows user  $u_j$ , then there exists a trust relationship between them, or simply, user  $u_i$  trusts user  $u_j$ . Trust network[1, 2] is an unweighted and undirected graph  $\text{TrustG}(U, \text{Tr})$ , where  $U$  represents the set of users and  $\text{Tr}$  represents the collection of edges between users.

The Trust network can also be represented in the form of a weighted graph where edge weight between user  $u_i$  and user  $u_j$  is set to 1 if  $u_i$  trusts  $u_j$ , else it is set to 0.

### 2.1.3 Graph Representation of Users

User set  $U$  is mapped onto a weighted graph  $G(U, E, W)$  [1, 2], where  $E$  represents set of edges linking users and  $W$  represents calculated similarities between different users in  $U$ . To calculate the edge weights between users, a combination of Pearson correlation coefficient and the trust network previously obtained is utilized [1], as given in the below formula:

$$w(u_i, u_j) = \alpha \cdot Tr(u_i, u_j) + (1 - \alpha) \cdot sim(u_i, u_j). \quad (3)$$

where  $Tr(u_i, u_j)$  denotes the trust score between users  $u_i$  and  $u_j$ .  $\alpha$  is the control parameter that adjusts contributions of trust  $Tr(u_i, u_j)$  and similarity  $sim(u_i, u_j)$  values.

### 2.1.4 User Clustering

User clustering is a technique used to group users with similar preferences in recommender systems. A graph clustering algorithm based on graph compression [3] to cluster users in a large and sparse user-food rating matrix. Nodes with high similarity are iteratively merged, and edge weights are updated according to node density and influence.

User clustering improves performance of user-based rating prediction by selecting more relevant neighbors for target user.

### 2.1.5 User-based Rating Prediction

Prediction for ratings of food  $f_k$  based on user  $u_i$  is calculated as given below [1]:

$$p_k^{u-based}(u_i) = \bar{r}(u_i) + \frac{\sum_{u_j \in C_i} w(u_i, u_j) \cdot (r_k(u_j) - \bar{r}(u_j))}{\sum_{u_j \in C_i} |w(u_i, u_j)|}. \quad (4)$$

where  $w(u_i, u_j)$  indicates weights between users  $u_i$  and  $u_j$ .  $C_i$  represents community to which user  $u_i$  belongs to.

Fig 2.2 [1] shows how user-based rating prediction is being done for a sample dataset with nine users [1]. User-food matrix containing rating for food given by user is used to find similarity between users. This, along with the trust network between users is used to build a graph representation of the users using which users are clustered. This is used to predict ratings of previously unseen food item with respect to each user.

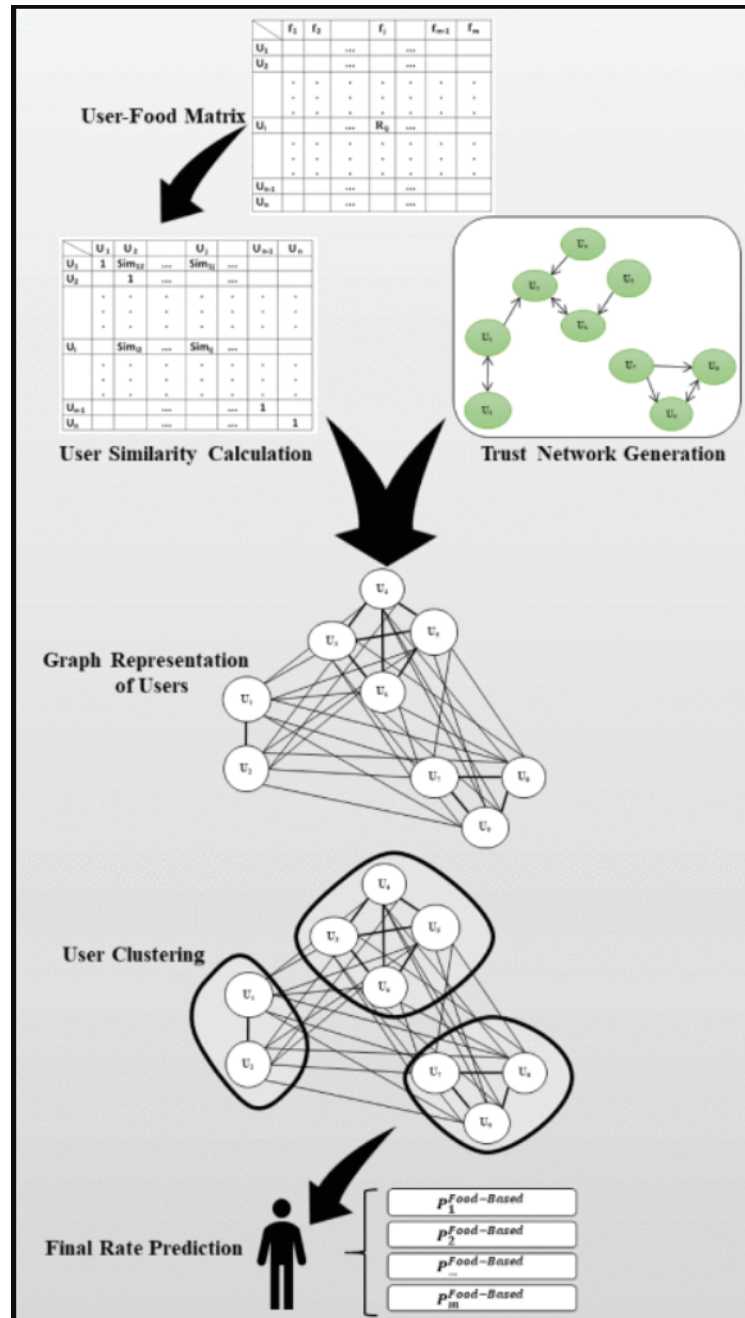


Figure 2.2: User-based rating prediction model for sample dataset with nine users

## 2.2 Food Based Rating Prediction

### 2.2.1 Food Deep Embedding

Food clustering based on ingredients is performed by utilizing the Bidirectional Encoder Representation from Transformer - Large (BERT-Large)[5]. It creates contextualized embeddings for foods. Inorder to apply Natural Language Processing(NLP) to food clustering, each food  $f_i$  is considered as a sentence, with its ingredients  $I_i$  as words[1, 2] of the sentence.

Fig 2.3[1, 2] shows how BERT is being used for food-based deep embedding[1]. To use Natural Language Processing (NLP) techniques for food clustering, each food  $f_i$  is treated like a sentence, while its ingredients  $I_i = \{ing_{\sigma(1)}, ing_{\sigma(2)}, ing_{\sigma(3)}, \dots, ing_{\sigma(k_i)}\}$  are treated as words in that sentence[1, 2]. Feature extraction procedure takes the sentences(i.e. foods) and tokens(i.e. ingredients) as inputs, and generates an output JSON file - which contains simulated embeddings from different layers of BERT[1, 5]. The embeddings are essentially representations of tokens as N-dimensional vectors, helping in capturing the context in which they appear.

The final step here involves calculating average of all token representations within the sentence inorder to produce a contextualized representation[1, 2].

### 2.2.2 Food Similarity Calculation

By utilizing contextualized embeddings from previous step, ingredients present in a food can be identified. Similar food items are in close proximity to each other in the vector space and they are assumed to share some ingredients[1, 2]. Euclidean distance formula is used in order to evaluate proximity between vectors of foods  $f_i$  and  $f_j$ .

Let  $f_i = \{f_{i1}, f_{i2}, f_{i3}, \dots, f_{iL}\}$  and  $f_j = \{f_{j1}, f_{j2}, f_{j3}, \dots, f_{jL}\}$  be contextualized representation vector of food  $f_i$  and  $f_j$ , respectively[1]. Similarity between  $f_i$  and  $f_j$  is calculated using Euclidean distance formula given as[1, 2]:

$$Sim(f_i, f_j) = 1 - \left( \sqrt{\sum_{l=1}^L (f_{il} - f_{jl})^2} \right). \quad (5)$$

where  $f_{il}$  denotes l - th dimension of the contextualized representations vector of the food  $f_i$ .

### 2.2.3 Food Clustering

This step is done to group foods based on their ingredients, and overcome the data sparsity and cold start problems of collaborative filtering-based food recommender systems. It reduces distance between similar embedding vectors.

Deep Embedded Clustering technique is used to reduce distance between similar embedding vectors in embedding space[6]. AutoEncoders and Kullback-Leiber divergence is used here to decrease the data dimensions and enhance the embedding vector representation[1, 2].

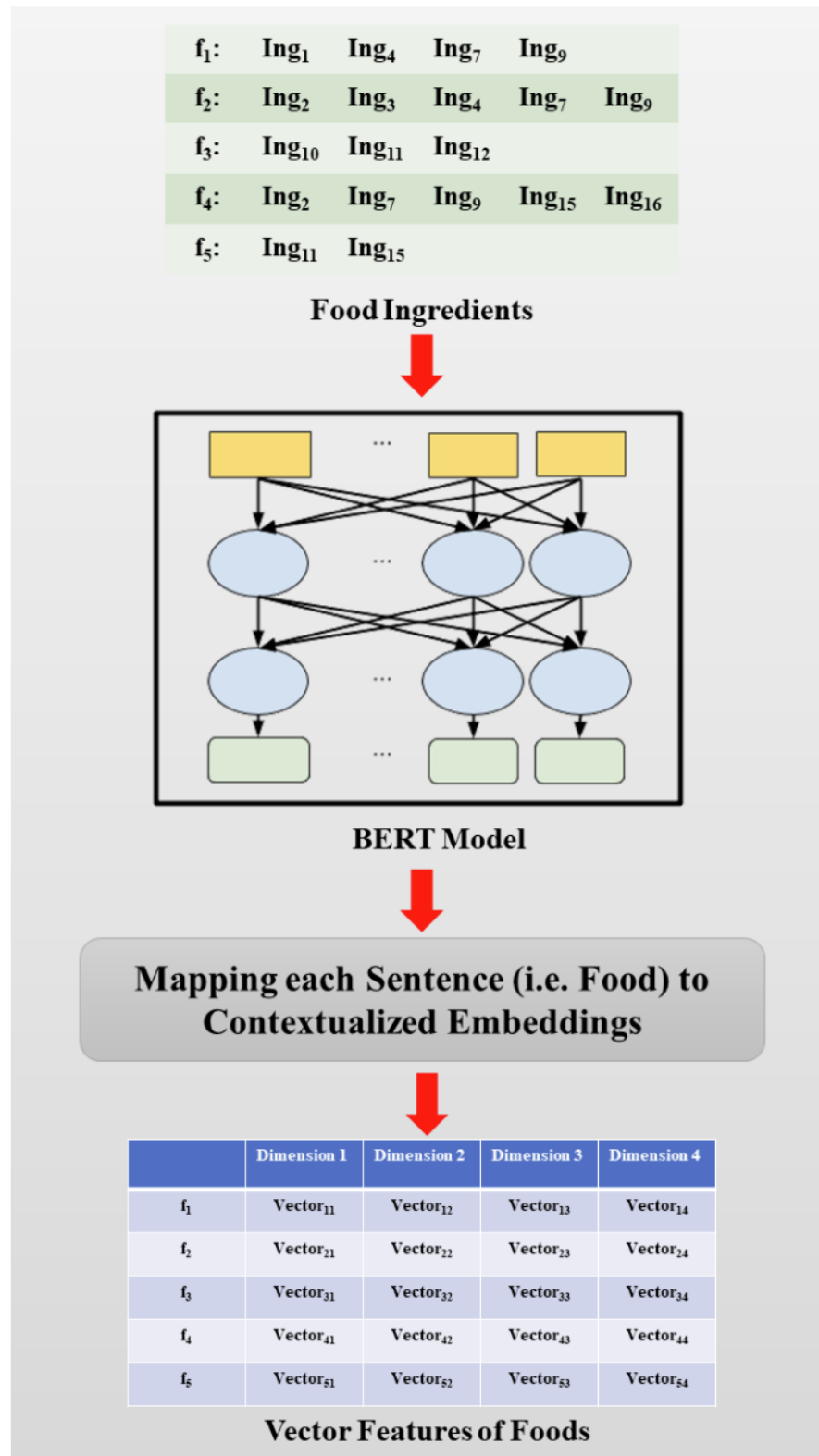


Figure 2.3: Overview of BERT-based food embedding method

### 2.2.4 Food-based Rating Prediction

For food-based rating prediction, rating of food  $f_i$  for user  $u$  is calculated as:

$$p_i^{f-based}(u) = \bar{r}_i + \frac{\sum_{j \in C_{f_i}} sim(f_i, f_j) \cdot (r_j(u) - \bar{r}_j)}{\sum_{j \in C_{f_i}} |sim(i, j)|}. \quad (6)$$

Fig 2.4[1, 2] shows food-based rating prediction for a sample dataset having seven different food items[1, 2]. User-food matrix and the food ingredient matrix is utilized along with BERT to perform food deep embedding. Proximity of food items in vector space is used as a factor in measuring similarity between them, and Euclidian distance formula is used to calculate this similarity. Deep Embedded Clustering technique is used to cluster the food items. Based on similarities between different foods and their previous ratings given by users, ratings of unseen foods are predicted.

## 2.3 Top-N Recommendation

After performing both user-based rating prediction and food-based rating prediction, the final prediction of food  $f_i$  for a user  $u$  is calculated as a convex combination of these predicted values[1, 2]. It is done by using the formula:

$$p_i(u) = \beta p_i^{u-based} + (1 - \beta) p_i^{f-based}. \quad (7)$$

where  $p_i^{u-based}(u)$  and  $p_i^{f-based}(u)$  denotes user-based prediction and food-based prediction on food  $f_i$  for user  $u$ , respectively[1, 2]. Parameter  $\beta$  controls whether user-based or food-based prediction value has a higher priority in final outcome. A higher value of  $\beta$  means user-based rating prediction was given a higher priority in influencing the final outcome, while a lower value means food-based rating prediction was given higher priority.

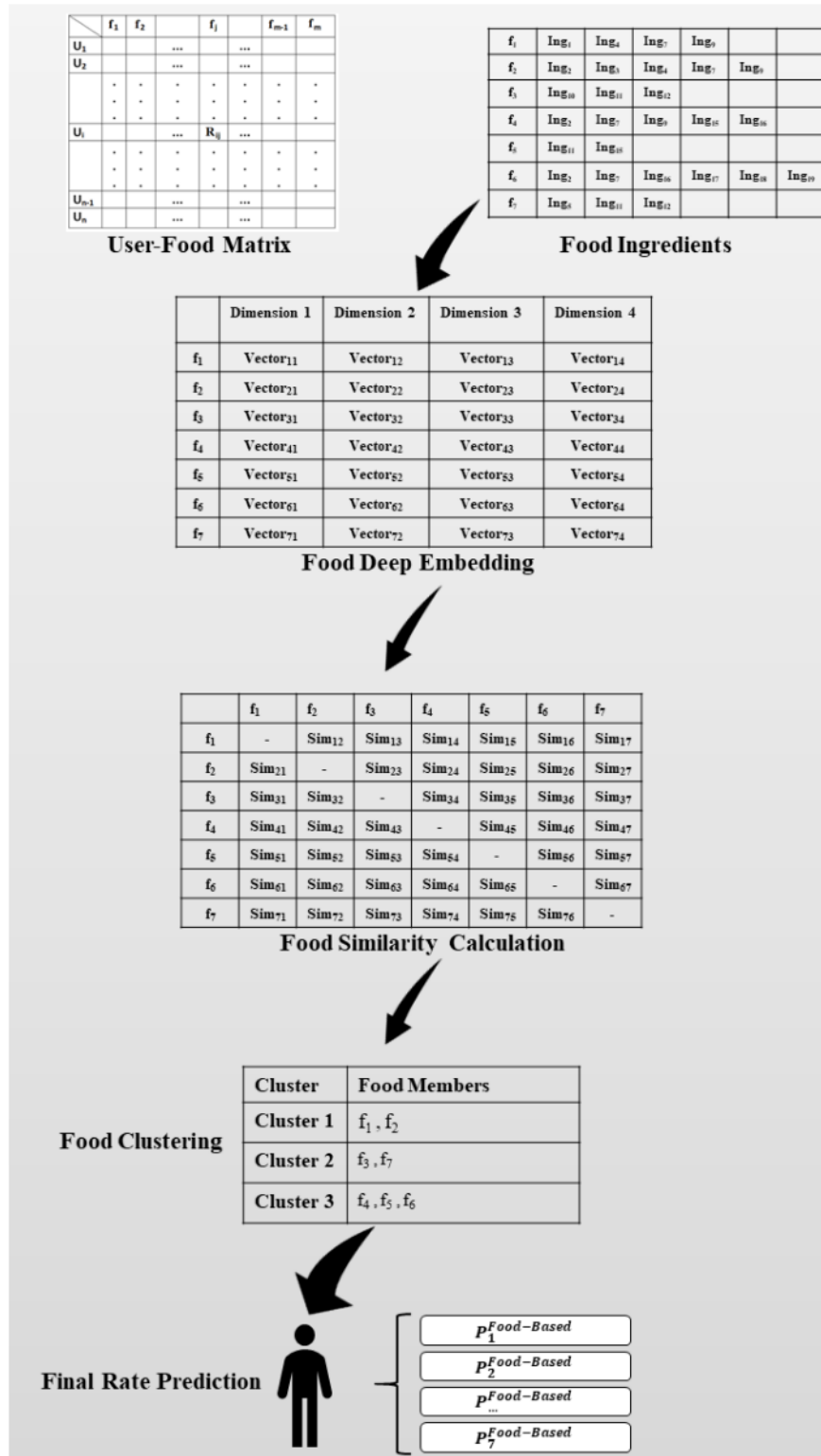


Figure 2.4: Food-based rating prediction model for sample dataset with seven users



## Chapter 3

# Conclusion

The development and implementation of a robust and efficient food recommender system is a timely and relevant endeavour in today's world. This is primarily due to the increasing health consciousness among people. As individuals strive for personalized dietary recommendations that align with their health goals, dietary restrictions and taste preferences, a system that can properly cater to these individuals is much needed.

The digital age has ushered in an era of information overload. The vast amount of food-related information widely available online can be overwhelming for many users. In such scenarios, recommender systems work as a valuable tool in helping the users to navigate through this plethora of information and discover new tools which align with their needs and goals.

Personalization is an important aspect which underscores the need for such a system. They significantly help in enhancing user satisfaction and engagement with the platform. This increases user retention and makes the system very beneficial for users.

In conclusion, TDLGC addresses several challenges faced by currently existing systems and meet the current needs and expectations of users for personalized and health-conscious food recommendations. It can greatly contribute to promotion of healthy eating habits of its users, thereby making it a valuable tool for the users.

# References

- [1] Mehrdad Rostami, Mourad Oussalah, and Vahid Farrahi. “A Novel Time-Aware Food Recommender System Based on Deep Learning and Graph Clustering”. In: *IEEE Access* 10 (2022), pp. 52508–52524. DOI: 10.1109/ACCESS.2022.3175317.
- [2] Mehrdad Rostami, Mourad Oussalah, and Vahid Farrahi. *OuluREPO Repository*. 2022. URL: <http://jultika.oulu.fi/Record/nbnfi-fe2022071551713>.
- [3] Xingwang Zhao, Jiye Liang, and Jie Wang. “A community detection algorithm based on graph compression for large-scale social networks”. In: *Information Sciences* 551 (2021), pp. 358–372. ISSN: 0020-0255. DOI: <https://doi.org/10.1016/j.ins.2020.10.057>.
- [4] Parham Moradi, Sajad Ahmadian, and Fardin Akhlaghian. “An effective trust-based recommendation method using a novel graph clustering algorithm”. In: *Physica A: Statistical Mechanics and its Applications* 436 (2015), pp. 462–481. ISSN: 0378-4371. DOI: <https://doi.org/10.1016/j.physa.2015.05.008>.
- [5] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].
- [6] Junyuan Xie, Ross B. Girshick, and Ali Farhadi. “Unsupervised Deep Embedding for Clustering Analysis”. In: *CoRR* abs/1511.06335 (2015). arXiv: 1511.06335. URL: <http://arxiv.org/abs/1511.06335>.