2023 3rd International Conference on Intelligent Technologies (CONIT)
Karnataka, India. June 23-25, 2023

Food Recommendation System Based on Collaborative Filtering and Taste Profiling

Kaustubh Sharma
Department of Computer Science and Engineering
PES UNIVERSITY
Bengaluru, India
officialkaustubh@gmail.com

Koushik Varma Mandapatti
Department of Computer Science and Engineering
PES UNIVERSITY
Bengaluru, India
mkoushikvarma@gmail.com

Malhar Chandrakanth Pattekar
Department of Computer Science and Engineering
PES UNIVERSITY
Bengaluru, India
pattekar.malhar01@gmail.com

K. K. Tarun Kumar
Department of Computer Science and Engineering
PES UNIVERSITY
Bengaluru, India
tarunkumar10020011@gmail.com

Gowri Srinivasa
Department of Computer Science and Engineering
PES UNIVERSITY
Bengaluru, India
gsrinivasa@pes.edu

***Abstract*— With restaurant menus becoming vast and dish names complicated, we present a recommendation system to facilitate ordering food at restaurants. The design approach is three-pronged and comprises the following options: (I) recommendations for you: menu items that are custom selected for a user through collaborative filtering based on the rating of dishes previously ordered and a targeted model that selects items based on a detailed comparison of reviews, such as any mention of ingredients, cooking technique, etc., where such reviews are available, (ii) similar items: recommendations that provide alternatives based on the ingredients and similar method of preparation of items previously selected and (iii) frequently bought together: recommendations that target increasing the 'revenue per order' based on association rule mining using the apriori algorithm. All these models have been incorporated into an existing food-ordering platform – 'Instead' – developed by the team and beta-tested at multiple locations. The recommendation system presented in this work will enable a user to walk into any restaurant and order an item consistent with their taste palette.**

Keywords—Food recommendation, taste profile, collaborative filtering, association rule mining, natural language processing

I. INTRODUCTION

Recommender systems are tools that facilitate narrowing down options by filtering the information available based on users' preferences or needs. Recommender systems help individuals understand content relevant or of interest to them from an overwhelming amount of information. Such a recommendation is often based on the opinions of communities of users. In short, a recommender system is a software, which processes a large volume of complex data and is expected to provide items that are relevant to the user. Recommender systems are used in many domains; they are particularly useful for recommending products to consumers and for targeted advertisements to get word of the product out to relevant user groups. [1-3]

A food recommendation system is a software that uses a combination of algorithms, data analytics, and machine learning techniques to provide personalized recommendations for what types of food a user might enjoy. The system typically collects and analyses data on the user's past orders, reviews, ratings, and other behavioural data to understand their food preferences, dietary restrictions, and other relevant information. It may also consider broader factors such as the user's location, time of day, weather, and popular trends. The system uses this data to generate a set of recommendations that are tailored to the user's preferences and needs. These recommendations maybe based on similar users' preferences, popular dishes in the user's area, or other relevant factors. Some food recommendation systems may also use natural language processing and sentiment analysis to extract insights from user reviews and feedback. This can help the system to identify specific foods, ingredients, or cuisines that users particularly enjoy or dislike. Overall, a food recommendation system is designed to help users discover new and interesting food options that they might not have otherwise considered, while also simplifying the process of finding and ordering food online.

It is human tendency to rely on recommendations either through reviews printed in papers, word of mouth or recommendation letters. These recommendations are especially helpful when people are forced to make a choice without sufficient personal experience of the alternatives. Recommender systems augment this natural social process with the help of smart algorithms specifically designed to assist a user. Multiple companies have made use of this human tendency in order to grow their businesses with the help of recommender systems.

Predicting dishes for a person is a difficult task as people crave different items at different times. Some of the many factors that influence a person's taste profile include their age, gender, genetics, nutritional knowledge [4], health status, willingness to try new dishes and culture. This project aims at building a model to make these recommendations as accurately as can be made with the data available. Further, these recommendations are designed to be made in real-time to users of an application designed by the team called – Instead.

II. APPLICATION 'INSTEAD'

'Instead' is an application designed and implemented by the team to help with placing orders in restaurants or food courts (see Fig. 1 for sample screenshots of the application). Designed during the pandemic, the application is ideally suited for college messes with multiple cuisines or food courts in offices, malls or hospitals or at carnivals or special gatherings. Basic details of the users (such as their age, etc.)

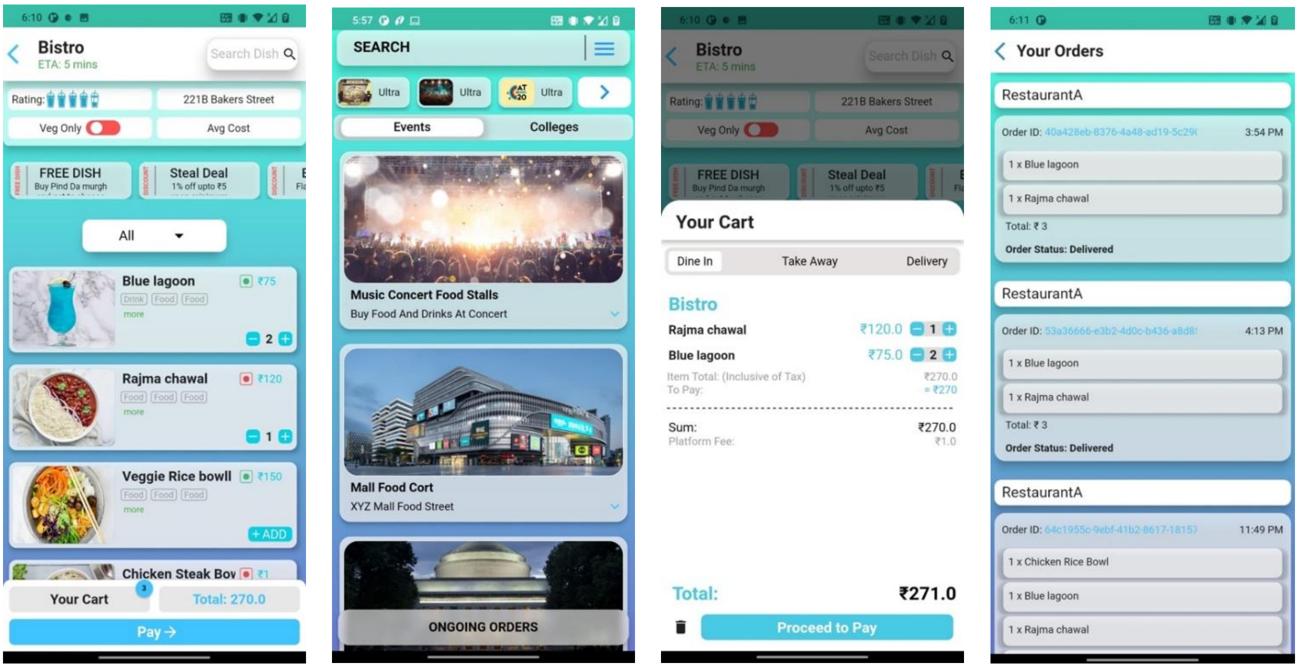

Fig. 1. Screenshots of the food ordering application 'Instead'.

are available at login and location is available when the application is in use. Restaurant owners are required to register with the application. The convenience of ordering food of their choice with not having to wait makes it attractive for users. Increased business and the system incorporating recommendations that boost revenue per order serves as an incentive for store owners. Thus, the incorporation of a recommendation system into 'Instead' is well motivated.

III. RELATED WORK

Customized recommendations have traditionally made use of user information such as user interest, previous ratings, age, gender and profession of the user, etc., for a more personalized experience [5-6]. Some examples highlighting the correlation between the foregoing parameters and preferences include teenagers or youngsters preferring spicy food and fast food versus senior citizens not ordering these types of foods as much. On the same vein, data seems to show a higher percentage of men prefer spicy food compared to women. Further, extrapolating these trends, one might find correlations between profession and exposure with the preferences: for example, a doctor might be more health conscious in their choice of foods. The profession of a user is also seen to have a significant effect on the time at which one eats and hence, the time the orders are placed by the user.

In the work by Patil et al., artificial neural networks (ANN) has been used to create recommendation scores for each dish on the restaurant's menu [5]. Initially, when there is less data for a particular user, the accuracy is low but with the user placing orders over time, the models learns the user's preferences and the accuracy of prediction increases. With this incremental model, the weights of the ANN represent the user's taste profile. Future work may involve adding nutritional information about the dishes, inclusion of other factors that play a role in determining a user's preference such as the price of the item and the nutritional value of the item that the user normally purchases, etc. [7]

Aghdam et al., focus on using non-negative matrix factorization (NMF) to collaborative filtering to overcome limitations of dealing with scale and sparsity of traditional collaborative filtering models [8]. Matrix factorization uses latent factors, such as features that explain the behaviour of users to estimate the rating for a particular user on an item, given a matrix of all the known ratings of various items by various users [9]. The cost function used for this NMF model is the generalized Kullback-Leibler divergence. One drawback of this NMF model is that the results can differ significantly with changes to the initialization of the matrix. A model trained on a dataset of movies consisting of ratings of over 1600 movies from over 900 users with five-fold cross-validation achieved a mean absolute error (MAE) of 0.9096 across all five folds and an average RMSE of 1.1994.

Another version of collaborative filtering based on similarity between user preferences is the recommendation of an item to a target user from the preferences of the k most similar neighbours [10-11]. In sparse data conditions, the accuracy of a suggestion made by collaborative filtering is seen to drop drastically due to the limitations of similarity metrics adequately capturing meaningful patterns [12-13].

Evolutionary collaborative filtering techniques can be used to reduce or eliminate constraints of collaborative filtering such as limited content analysis and overspecialization [14]. The apriori algorithm is used to create user profiles based on the ratings and categorical features of items. The dataset utilized to demonstrate the utility of this approach is the MovieLens Dataset (one of the highly used datasets for testing ML models) and the efficiency of the proposed recommendation is evaluated to reveal association rule mining far surpasses collaborative filtering [15-16].

To improve the performance of a recommendation system there are multiple algorithms but two primarily used algorithms are collaborative filtering (CF) and content-based filtering (CBF) [17-18]. The CB filtering technique is primarily based on the information provided by users and

2

objects. A CB Recommendation-System suggests an item to a user based on the user's profile, the item's profile, user reviews, and other factors. CF suggests a product to a user based on their previous ratings. The two methodologies utilized in the CF-based Recommendation-System are model-based CF and memory-based CF. Model-based CF demands the use of machine learning techniques such as clustering, classification, and rule-based systems, among others. In memory-based CF, two important processes are user/item similarity computation and rating prediction [14]. Various similarity measures (SMs) have been proposed in the literature to evaluate the similarities between users/items. We adopt the apriori approach since most classic similarity measures, such as cosine similarity, are unable to estimate similarity when user ratings are not co-rated. Furthermore, most traditional similarity metrics, such as Equal-Ratio, Unequal-Length, Flat-Value, Opposite-Value, Single-Value, and Cross-Value, have flaws. By overcoming the constraints of existing similarity metrics, an optimal collaborative filtering-based recommender system is provided.

The study demonstrates that after profiling, the system can generate highly personalised and optimized recommendations. Based on the specified rating, the gathered datasets from 19 genres are categorized into 2 sets: liking and disliking categories. After that, the Apriori algorithm was used to determine the users' final preference list or user profile. These characteristics are then utilized to compute similarity measures and rating prediction techniques [10]. They state that CF-based RS has better precision-recall, F1-measure, and accuracy than existing CF-algorithms that use Jaccard Similarity, Cosine Distance, and other measures. Under the different sparse scenarios, all conventional similarity measures boost the rating prediction accuracy using our proposed technique, concluding that user profiling has shown significantly superior accuracy than older existing methods.

The paper aims at recommending food recipes to tackle the obesity epidemic by concentrating on the two initial dimensions of food recommendations, data capture and food recipe relationships [7]. When adapting to a healthy lifestyle most people struggle due to the lack of information on sustaining a healthier lifestyle, the paper aims at helping users by recommending a healthy recipe to them based on their personal liking. The paper compares the performance of a content-based filtering approach, a collaborative filtering approach and a hybrid model of the two aforementioned models. The system made in the given paper aims at making recommendations based on the ratings given by users along with the food items used in the recipe. The system makes use of two data-gathering strategies: the first is a fine-grained food item strategy that gathers explicit ratings on individual food items, that is, all weights, methods of cooking and combinations are ignored and considers all items with equal weights. The second strategy is a higher-level strategy that gathers ratings on recipes. The hybrid model makes use of collaborative filtering so as to deal with the problem of a sparse matrix, it makes use of Pearson’s correlation so as to find the N nearest neighbours in order to predict the ratings given by users for recipes that they have not rated yet [19]. The system uses the newly generated data for content-based filtering. Similar models were built using the second strategy where importance was given to the recipe rather than just the food ingredients used in the making of the dish [20].

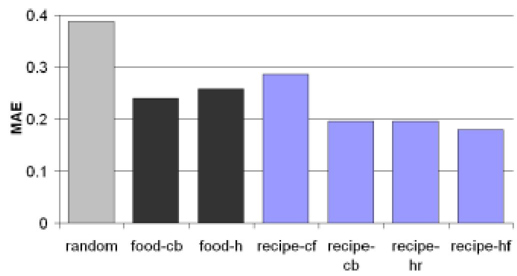

Fig. 2. Performance of various strategies

The performance of the various strategies used is depicted in Fig. 2, which clearly shows that the hybrid model shows the least Mean Average Error (MAE) and is thus the best model from the above approaches [21].

IV. PROPOSED METHODOLOGY

In order to make a recommendation engine which most accurately identifies and matches user taste profiles to each other, we require to match users based on more than just the ratings, ingredients, reviews, etc. The proposed approach to deal with our problem statement is to make use of a complex model, which will make use of the outputs from multiple models and thus recommend dishes to the user most accurately. Our proposed recommendation system consists of 3 different sub-modules each having a particular characteristic which would ultimately help in the final model. They are as follows:

A. Recommended for you

There are items that are handpicked by the models for the user, based on their previous orders and taste profile as computed by the algorithms. This further consists of two recommendation models, The first one being the collaborative filtering model that takes into account various ratings given by various users to the dishes. Based on these ratings a similarity score is computed between the various users and recommendations are made based on the data of users who are similar to the user. Another model that selects similar users is a user similarity review model that we have developed that matches the users based on the similarities in their reviews. This model compares the reviews in detail to pick out various subjects a user is talking about in a review and computes the sentiment for that subject.

For example. If the user says “I love the brinjal and I don’t like the sauce.” The model processes this sentence identifying parts of speech and delineating two subjects which are the chicken and the sauce and then further run sentiment analysis on them to tell us, [Brinjal - Positive, Sauce - Negative]. Then we go out and try to find other users which have similar reviews which tells us their taste profiles are similar and hence similar items can be recommended to each other. These two models work in tandem to give the best possible results for each user, so that as soon as the user opens the menu of the restaurant, we can say with a high degree of confidence that the dishes listed at the beginning are the ones the user is most likely to order. If we have less amount of data for a particular user i.e., the cold start problem, in that case, we just display the items which are highly rated and most frequently ordered in that restaurant.

B. Similar Items:

When a user selects a particular item we want to show them other alternatives for that dish. These alternatives will taste mostly the same but will provide the user with some more choices for a recommended dish. These items will be displayed when a user adds a particular item to his cart. This model works on two underlying principles, comparing the ingredients as well as comparing the method of preparation. The principle logic is that dishes with the same ingredients and similar methods of preparation would largely taste the same. An intersection of the results from these two models will be what is displayed to the user.

C. Frequently bought-together items:

This model tracks frequently paired items and displays them to the user before the order is placed. This is shown to the users once all the items are chosen. Frequently paired items are calculated using the apriori algorithm, which looks at various orders and checks which are the items that users generally buy together. This model is meant to increase the metric of 'revenue per order', which will be a major selling point for the restaurants. The results of this model are not substitutes but complementary to the already selected items.

V. IMPLEMENTATION

Dataset: Finding data related to food orders placed online is difficult, as it is impossible to access as this is valuable data. To overcome the above problem the project makes use of a database of food recipes, published by Food.com. Each recipe is considered to be a food item available on the restaurant menus [22].

The data provided by them is available in multiple formats, allowing us to choose how and what kind of data is required. We make use of the following tables throughout this paper.

1. PP_recipes.csv
2. PP_users.csv

The reason for using the above-specified dataset is that it provides us with more details regarding individual dishes, such as the ingredients, method of preparation, the time needed to prepare the food item, the nutritional values of the dishes, etc. This data will help us in finding more similarities between dishes.

Along with dish details the dataset provides us with information about the users who make these dishes, which we consider as the user's previous orders. It provides us with the review given by each user regarding particular dishes, thus also allowing us the possibility of finding similarities between users based on their preferences, likes and dislikes.

As mentioned in the proposed methodologies, we have built multiple models which we believe will help in making recommendations as accurately as possible. These models make use of various user data in order to make the recommendation, including the food rating, reviews, ingredients etc.

For the "Recommended for you" model, we utilize 2 different machine learning algorithms. Initially, we perform collaborative filtering on the dishes and users in order to get the ratings given by various users. We make use of a cosine similarity matrix to find the similarity between the users based on the dishes that they have previously rated. This similarity between users is calculated using the following formula,

Sum the multiplication of ratings of 'i' and 'j' given by all users U

$$similarity(i,j) = \frac{\sum_u^U r_{(u,i)} r_{(u,j)}}{\sqrt{\sum_u^U r_{(u,i)}^2}\sqrt{\sum_u^U r_{(u,j)}^2}}$$

Multiplication of Square roots of sum of squared ratings of item 'i' and 'j' by user 'u'

Once the similarity between users is established, this similarity score is used to predict the rating a user is likely to give to a dish, and the recommendations are then made based on these predicted ratings.

The second algorithm utilized is an NLP model where we perform a sentimental analysis on the dataset in order to analyze the user reviews and feedback to determine the overall sentiment towards specific dishes or restaurants. First, we perform NLP on each individual dish review to look for specifics that a user might or might not have liked about a specific dish. To do so we extract nouns and subjects of the reviews left by the user and perform sentiment analysis to establish their sentiments towards those specifics. The sentiments are classified as either positive or negative. After classifying the reviews, we utilize them in order to find users who have had a similar sentiment towards the same details of the dish, and then make recommendations based on the likes and dislikes of these similar users [11].

The outputs of the above two mentioned algorithms are combined by taking an intersection of the outputs and are stored for later.

The second part of the recommendation engine is the "similar items" model. In this model, we draw out similar dishes based on their characteristics such as the ingredients used to prepare that dish and the method of preparation of the same. Our goal with this part of the recommendation engine was to analyze the user's liked dishes and to use this information to predict/recommend dishes they are most probably going to like.

The first part of this model uses the ingredients of the dish to figure out the similarity between dishes. In order to calculate this we first formatted the known ingredients from the database into a usable format. The ingredients are processed and split into tokens, later used to find the similarity by making use of the bag of words technique. We explored a handful of similarity metrics such as TF-IDF and Jaccard similarity, but upon further investigation, we realized that cosine similarity is the best match for our use case. So, once the similarity between dishes is found, we can have a rough understanding of how likely a user would like dish 'B' if he has liked dish 'A'. Using the determined similarity between dishes, we predict the rating a user would most likely give to each dish based on which we decide whether to recommend that dish to a particular user or not.

For the second part of the model, we analyze the method of preparation of the dishes. We do so by making use of NLP in order to find information specific to the method of preparation. We make use of word2vec so as to get the semantic meanings of the processes and then use this to find the similarity between dishes. Recommendations are made based on these similarities.

4

The final output of these models is combined by the intersection of the outputs generated by the “similar dishes based on ingredients” model and the “similar dishes based on the method of preparation” model and stored to use later.

The final part of the recommendation system, which is the “Frequently bought together” model makes use of the Apriori algorithm. The apriori algorithm operates on a dataset that consists of transactions, where each transaction is a set of items. The first step in the algorithm is to identify all the individual items in the dataset and calculate their support, which is defined as the percentage of transactions that contain a particular item. Items with a support value greater than a predefined threshold are considered frequent items.

The second step of the algorithm is to generate candidate itemsets of size two by joining frequent itemsets of size one. The candidate itemsets are then pruned based on their support. Any candidate itemset with support less than the predefined threshold are discarded. This process is repeated to generate candidate itemsets of increasing size until no more frequent itemsets can be generated.

The third and final step of the algorithm is to generate association rules from the frequent itemsets. An association rule is a statement of the form "if item A is present, then item B is likely to be present as well". The confidence of an association rule is defined as the percentage of transactions that contain both items A and B out of the transactions that contain item A. Association rules with confidence values greater than a predefined threshold are considered strong rules.

The outputs from all the above-mentioned models is combined with preference given to the dishes that have been outputted most frequently by the various models. This combined output is then recommended to the users.

VI. RESULTS AND DISCUSSION

As a measure of the precision of each of the models, we calculated the ratio between the number of dishes ordered by the user out of all the recommendations made to them and the number of orders made by the user. This measure was compared across all the models and used to combine the results of all three models.

The precision of the models is the average of the precision of the recommendations made to fifteen random users, who have made at least five different orders. These users are kept constant across all the models so as to give us the most accurate comparison between the models.

We built many models over the course of this project and have made significant improvements from where we started, to where our final system now stands. The precision of our very first model, which was the linear regression model was found to be very low at only 18%, but with changes in the models used, modifications made to the data and improvements made to the technique, we have gotten the precision of the final model to be 85%.

TABLE I. MODEL PERFORMANCE FOR VARIOUS MODELS.

Model	Precision
Final Model	0.85
Sentiment Analysis Model	0.8
Collaborative Model	0.65
Linear Regression Model	0.18

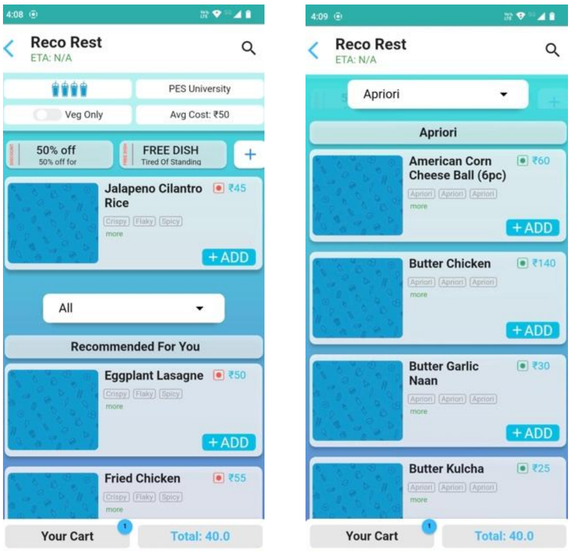

Fig. 3. Recommendations made to a user on the food ordering application ‘Instead’

The recommendations made in the above images are made for a user who has a history of ordering spicy chicken dishes. As seen, the recommendations made by our model suggests spicy, non-vegetarian dishes to the user.

VII. CONCLUSION AND FUTURE WORKS

We have demonstrated the design and utility of a recommender system that takes into account users’ preferences and developing a taste profile to augment a collaborative filtering based model. Our results demonstrate found that the models performed reasonably well for most of the users, however, for users for whom there was very limited data, the models did not work as well. This is a problem not only for the present application, but any recommender system; it is well known that the more data available, the more will be the accuracy of the model.

We have reported the results on publicly available data since the data collected from our application ‘Instead’ was limited to the test deployment. There are parameters not available in the publicly available datasets that would be valuable in making meaningful recommendations, such as the price of a dish, the time of the day a dish is usually ordered, etc., which would be available to us through Instead.

A natural ‘next step’ includes leveraging the foregoing parameters from our application in building a more robust recommender system. Another aspect of augmentation would include experimenting with different semantic analysis techniques to mine patterns regarding the methods of preparations and the reviews given to the dishes.

REFERENCES

[1] Rishabh Mall, “Recommender System.” towardsdatascience.com. https://towardsdatascience.com/recommender-system-a1e4595fc0f0 (accessed Feb. 8, 2022).

[2] Gastón Rodríguez, “Introduction to Recommender Systems.” tryolabs.com. https://tryolabs.com/blog/introduction-to-recommender-systems (accessed Feb. 15, 2022).

[3] Luís Rita, “Building a Food Recommendation System.” towardsdatascience.com. https://towardsdatascience.com/building-a-food-recommendation-system-90788f78691a (accessed Feb. 4, 2022).

[4] M. Harvey, B. Ludwig, and D. Elswiler, “You are what you eat: Learning user tastes for rating prediction.” in Lecture Notes in

5

Computer Science, *String Processing and Information Retrieval - 20th International Symposium, {SPIRE}*, October 2013.

[5] P. D. Patil, N. P. Patil, P. Y. Geete, A. V. Nikat, P. S. Kulkarni, "A Survey on Existing Food Recommendation Systems." *International Journal of Science Technology & Engineering* (*IJSTE*), vol.3, pp. 15-19, June 2017.

[6] B. P. Majumder, S. Li, J. Ni, and J. McAuley, "Generating personalized recipes from historical user preferences." *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5976-5982, November 2019.

[7] R. Y. Toledo, A. A. Alzahrani, and L. Martínez, "A Food Recommender System Considering Nutritional Information and User Preferences." *IEEE Access*, vol. 7, pp. 96695-96711, 2019.

[8] M. H. Aghdam, M. Analoui, P. Kabiri "Application of nonnegative matrix factorization in recommender systems." *6th International Symposium on Telecommunications (IST)*, pp. 873-876, 2012.

[9] Y. Kumar & N. Kumari, "Movie Recommendation System." *Journal of Interdisciplinary Cycle Research*, vol. 12, issue 9, pp. 847-851, September 2020.

[10] N. Kadale, S. Lath, and S. Shinde, "Wireless Customizable Food Recommendation System Using Apriori and K-Means Algorithm." *International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS)*, Volume IV, Issue X, October 2015.

[11] M. O. Yadav, K. Mishra, D. Patil, E. Braganza, and C. Finny, "Design and Implementation of Movie Recommendation System based on NLP And Content-based Filtering algorithm." *Proceedings of the Fifth International Conference on Communication and Electronics Systems* (ICCES), 2020.

[12] S. Sawant and G. Pai, "Yelp Food Recommendation System." Standard University, 2013.

[13] T. Puraram, P. Chaovalit, A. Peethong, P. Tiyannunti, S. Charoensiriwath, and W. Kimpan, "Thai Food Recommendation System using Hybrid of Particle Swarm Optimization and K-Means Algorithm." *6th International Conference on Machine Learning Technologies (ICMLT)*, pp. 90-95, April 2021.

[14] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Scalable collaborative filtering approaches for large recommender systems." *The Journal of Machine Learning Research*, vol. 10, pp. 623-656, 2009.

[15] U. Kuzelewska, "Clustering Algorithms in Hybrid Recommender System on MovieLens Data." *Studies in Logic, Grammar and Rhetoric*, vol. 37, no. 1, pp. 125-139, 2014.

[16] P. K. Singh, E. Othman, R. Ahmed, A. Mahmood, H. Dhahri, and P. Choudhury, "Optimized recommendations by user profiling using apriori algorithm." *Applied Soft Computing Journal* 106, 2021.

[17] A. A. Fakhri, Z. K. A. Baizal, and E. B. Setiawan, "Restaurant recommender system using user-based collaborative filtering approach: a case study at Bandung Raya Region." *Journal of Physics: Conference Series*, vol. 1192, no. 1, 2019.

[18] D. Gunawardena and K. Sarathchandra, "BestDish: A Digital Menu and Food Item Recommendation System for Restaurants in the Hotel Sector." in 2020 *International Conference on Image Processing and Robotics (ICIP)*, 2020.

[19] X. Mao, Y. Rao and Q. Li, "Recipe popularity prediction based on the analysis of social reviews." 2013 *International Joint Conference on Awareness Science and Technology & Ubi-Media Computing (iCAST 2013 & UMEDIA 2013)*, Aizu-Wakamatsu, Japan, 2013, pp. 568-573, doi: 10.1109/ICAwST.2013.6765504.

[20] C.-Y. Teng, Y.-R. Lin, and L. Adamic, "Recipe recommendation using ingredient networks." *In Proceedings of the 4th Annual ACM Web Science Conference (WebSci '12)*, pp 298-307, June 2012.

[21] J. Freyne and S. Berkovsky, "Intelligent food planning: personalized recipe recommendation." February 2010.

[22] S. Hensley, "CS 229 Final Report: Recipe Rating Prediction (Natural Language)." *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.