Restaurant Recommendation System

Thanmayee Matha
Indraprastha Institute Of Technology
Delhi
thanmayee22084@iiitd.ac.in

Saksham Nautiyal
Indraprastha Institute Of Technology
Delhi
saksham22061@iiitd.ac.in

Abhinav Garg
Indraprastha Institute Of Technology
Delhi
abhinav22002@iiitd.ac.in

Komaljyot Kaur
Indraprastha Institute Of Technology
Delhi
komaljyot22105@iiitd.ac.in

Riya Garg
Indraprastha Institute Of Technology
Delhi
riya22058@iiitd.ac.in

Abstract

Every day, a new restaurant opens, contributing to the industry's fast expansion. Large e-commerce websites now heavily rely on recommendation systems. In order to assist consumers identify the finest restaurants that meet their interests, a recommendation system is required. For the purpose of giving consumers the finest restaurant suggestions, we want to create a hybrid restaurant recommendation system that blends personalized and non-personalized recommendations.

1 Introduction

Recommendation systems are widely used today. Almost every major digital business has used them in one way or another. For example, YouTube uses them to choose which video to play next on autoplay, while Amazon uses them to propose things to customers.

Design and implement a restaurant recommendation system by considering the user's past dining history and ratings, as well as the features of restaurants, such as location, cuisine, price range, ambiance, and ratings.

The system uses hybrid filtering which combines results of collaborative filtering to identify similar users based on their past dining history and ratings, recommend restaurants that the user might like and content based filtering techniques to recommend restaurants that match the user's preferences based on the features of the restaurant such as location, cuisine, price range, ambiance, and ratings to provide personalized restaurant recommendations to users. The system also provides a user-friendly interface that allows users to easily search for and discover new restaurants based on their preferences and recommendations.

The system is developed based on user reviews in Yelp Data-set which consists of various information about the restaurants including their menus, ratings, and reviews, as well as user preferences, including their cuisine preferences, price range, and other factors.

2 Motivation

Conventional means of locating restaurants, including asking around or pursuing internet review sites, can take a lot of time and may not offer choices that are specifically tailored to the user's tastes and region. Users may experience dissatisfaction and decision fatigue as a result of this. The main people who are going to benefit from this recommendation system are the tourists, who are new to a city. Most of tourists always love to visit famous restaurants in a particular city during their visit. So this motivated us to create a restaurant recommendation system.

3 Literature Review

By using the users' current geographical position, Md. Ahsan Habib et al. present a novel location, preference, and time based restaurant recommendation system. The method evaluates user check-in accounts to investigate his visiting habits, meal preferences, and popular eateries. Four main factors used to calculate recommendation scores are User preference score, Restaurants' separation, the hour of the day , and the popularity ratings of the eatery[1].

In order to predict the customer satisfaction rating, Nanthaphat Koetphrom et al. present a method based on real data (connected to customer/restaurant features) and similarity in consumer preferences. The three filtering strategies being suggested are collaborative, content-based, and hybrid. The results show that the collaborative filtering approach uses cluster-based methodology to regulate information among its peers[2].

The purpose of the study, according to Khushbu Jalan et al., is to recommend inn names to the explorer based on their interests and inclinations, using the feedback from other explorers and the rating as an incentive to increase prediction accuracy. The setting-aware cross-breed methodology is employed where CF method aggregates wistful research and provides personalized inn ideas. The use of a setting-based procedure then improves the proposal's results even more[3].

The adaptive climate is used in the café recommender framework developed by Jun Zeng et al. The framework eventually gives suggestion outcomes based on the model after first building an inclination model for customers based on client/café area nuances and café visits. The contextual study also showed that the BMCS and BWCS-based café recommender system could effectively use the client's tendency[4].

Ling Li et al. suggest three changes to the conventional UCF algorithm. The UCF algorithms' accuracy was quite poor because there were numerous factors that could affect a user's preference for a restaurant. Last but not least, actual private information of registered internet users is being used to gauge the similarity connected to user features. The results make it abundantly evident that the ACF-modified algorithm improves the accuracy of similarity computation and provides the user with a highly accurate restaurant suggestion[5].

Poriya et al. explores two types of recommender systems: Non-Personalized Recommender Systems (NPRS) and User-based Collaborative Recommender Systems (UCRS). While UCRS bases its recommendations on user behavior, NPRS makes suggestions based on an item's popularity. The research contrasts the benefits and drawbacks of the two approaches and proposes a hybrid strategy combining the advantages to increase suggestion accuracy. The specific needs of the application determine which system should be used[6].

Pavate et al. describes a methodology to recommend and classify cuisines based on reviews using supervised learning.The author trains a classifier to predict the cuisine type using information extracted from a dataset of Yelp reviews. A system for recommending comparable cuisines based on user preferences is also presented in the article. In the classification job, the author reports an accuracy of 84.5% and encouraging outcomes in the recommendation system. The suggested method can be helpful for systems that recommend restaurants and specific dishes[7].

Gupta and Singh et al. proposes a personalized restaurant recommendation system that uses location data and user preferences to suggest restaurants. The algorithm uses collaborative filtering to produce recommendations and considers factors like restaurant popularity, distance, and cuisine. On a dataset of Yelp reviews, the author ran experiments and contrasted the suggested system with various recommendation methods. The findings demonstrate that the proposed methodology outperforms competing approaches and offers more precise and individualized recommendations. Users looking for eateries in new areas or wishing to sample different foods may find the system helpful[8].

4 Dataset

The data we used comes from the following link:
http://www.yelp.com/dataset_challenge.
This data has been made available by Yelp for the purpose of the Yelp Dataset Challenge. In particular, the challenge dataset contains the following data:

- 1.6M reviews and 500K tips by 366K users for 61K businesses
- 481K business attributes, e.g., hours, parking availability, ambience.
- Social network of 366K users for a total of 2.9M social edges.
- Aggregated check-ins over time for each of the 61K businesses

4.1 Exploratory Data Analysis

The following are the findings that we reached after examining the Yelp dataset:

- We found that McDonald's and Subway are the most popular restaurants in all states.(Figure 1)

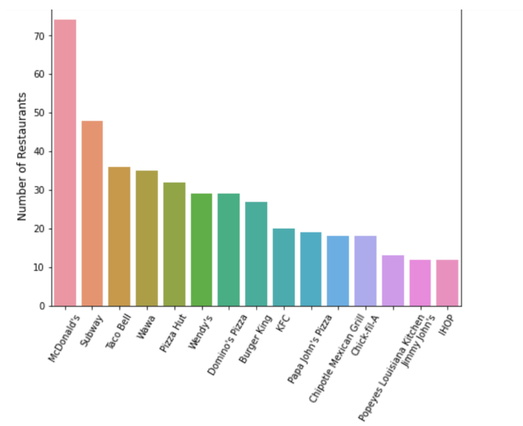


Figure 1:

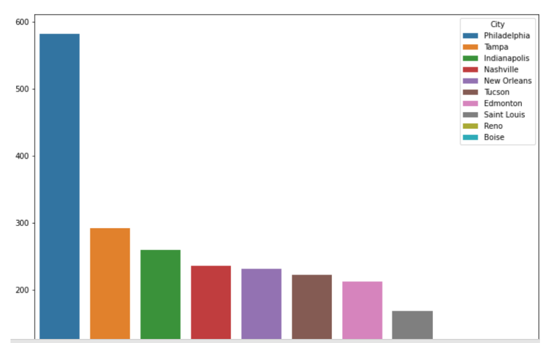


Figure 2:

- City Philadelphia has the maximum number of restaurants.(Figure 2)
- We see that locations of businesses are concentrated in clusters. These clusters must be big cities.(Figure 3)

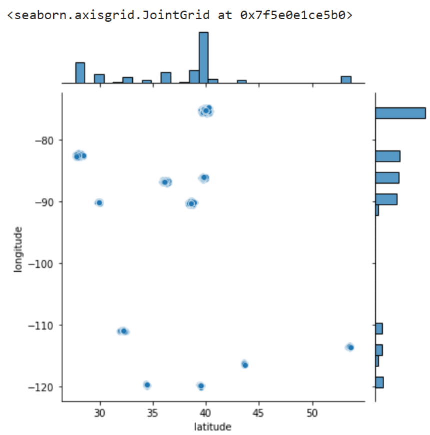


Figure 3:

- We analyzed that our data has businesses from certain cities of U.S. and not all over U.S(Figure 4)

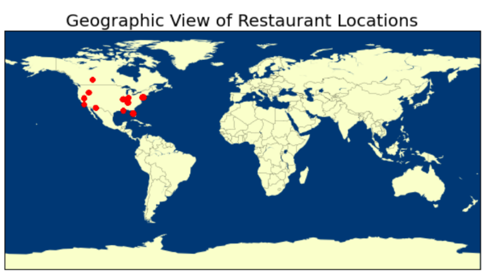


Figure 4:

- Then we explored most reviewed food categories in business data(Figure 5)
- Checked how rating and reviews are related to each other as these are important factors for restaurant recommendation.We can see that as

Figure 5:

the rating increases from 1.0 to 4.0, the number of reviews tends to increases as well. However, as rating increases further, especially from 4.5 to 5.0, the number of review shrinks.(Figure 6)

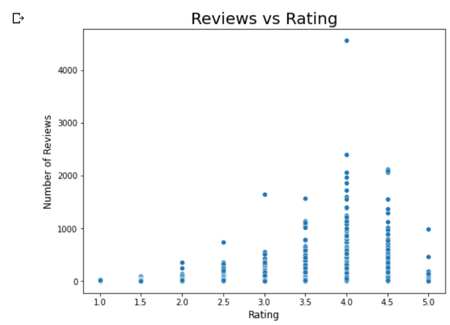


Figure 6:

- We also found the top 10 5-star restaurants sorted by review count(Figure 7)

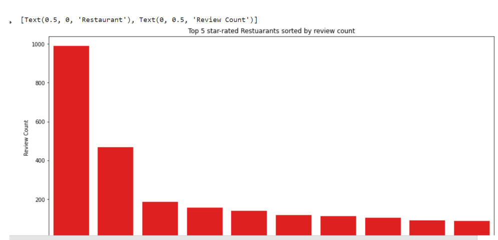


Figure 7:

- Restaurants with TakeOut, AcceptCreditCard, GoodForKids, Reservation, GoodForGroups, BusinessParking, HasTV, Alcohol, BikeParking, Delivery,, Attire are more popular

- We saw correlations roughly at the center of heatmap in business data
- We also analyzed the number of reviews in a year and the most used time to give reviews.(Figure 8 and 9)

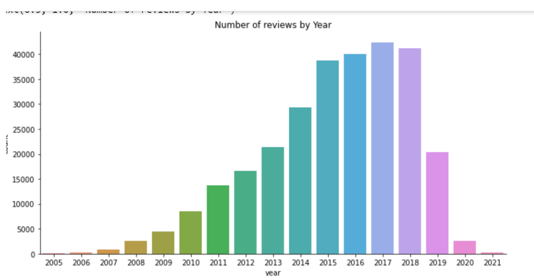


Figure 8:

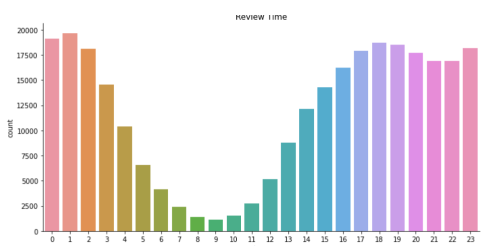


Figure 9:

- We found that no reviews recorded are minimum in the morning.
- Overtime users become less harsh reviewers.
- Half of the restaurants have less than 20 check ins, even less than the reviews, indicating that check in is not a widely used feature on Yelp.
- The popularity of reviews shows a steady upward trend since the beginning in 2004 with seasonal fluctuations, whereas the popularity of tips (No. of tips) increases in the first four years after its introduction (2009-2013) and slowly dives down thereafter. Therefore tips are not as popular as reviews.(Figure 10)

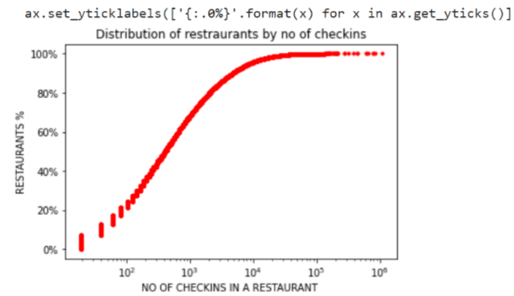


Figure 10:

5 Methodology

For this project, we have used the Yelp dataset as stated above.We also separated the data of the city Philadelphia to build its separate recommendation as this city is the most popular in the given dataset.
Initial preprocessing of the dataset consists of converting the json files into csv format and then cleaning them by removing unnecessary information.Then exploratory data analysis was performed as stated in above section.
For the baseline results We build a recommender system on item similarity based collaborative filtering. It recommends restaurants like the given restaurant id. For this we have used a KNN based model.
We also build a model to recommend the rating of the user so that we can get adjusted score later used in non personalized module .The baseline model has the average mean ratings of all the users.
We developed a hybrid recommendation system formed by non- personalization recommendation based on restaurant location and feature based keyword filtering, collaborative filtering based recommendation and personalized restaurant content based filtering recommendation module which is implemented using a user-friendly interface created to integrate the submodules, gather user interests and navigate users through the hybrid recommendation engine.
The non- personalization recommendation based on restaurant location and feature based keyword filtering facilitates combining restaurant feature-based

(cuisine, style, pricing) and location-based (zip code, city, state) keyword filtering of the restaurant catalog. The filtered catalog is ranked according to the user's preferred ranking criteria to produce the tailored suggestions.
To find the distance between a restaurant and the user's point of interest, it is essential to construct a function for computing the geodesic distances between two locations on a globe given their coordinates.We concluded that the optimum approach for this was to use the great circle distance calculated using a spherical earth model.
The original restaurant average star rating is replaced with an enhanced statistic called an adjusted rating score.

$$\text{score}_i = \frac{\sum_u r_{ui} + k * \mu}{n_i + k}$$

In the collaborative filtering-based recommendation module, personalized restaurant recommendations are supported given the distinct user id. The user-unrated restaurants from the catalog are ranked by the ratings anticipated by the model and returned as personalized suggestions. The personalization is computed based on the user's and all other users' rating histories of all Yelp establishments.
SVD, NMF, and other matrix factorization models are used as the initial step in implementation. The collaborative recommender feature is implemented using the optimised SVD with bias model since it provides the best RMSE for rating prediction on an unknown testset. The user latent feature matrix and bias vector, the business latent feature and bias vector, along with other relevant details of the trained best model are saved to file for usage . The best matrix factorization model is trained using the whole dataset prior to implementation.

In the personalized restaurant content-based recommender module, personalized restaurant recommendations are supported given the distinct user id. User-unrated restaurants from the catalog are ranked by similarity score and returned as personalized recommendations. The personalization is computed based on the similarity between the user's preference indicated by historical ratings and all restaurants' features extracted from a rich set of Yelp restaurant review texts.
Based on the cosine similarity between the restaurant and user feature vectors, we computed restaurant and user feature vectors, and a similarity score was created. According to similarity ratings, we ordered open restaurants in descending order.
Further baseline models and some updation used to build a seperate Philadephia city restaurant recommendation system which also was implemented.

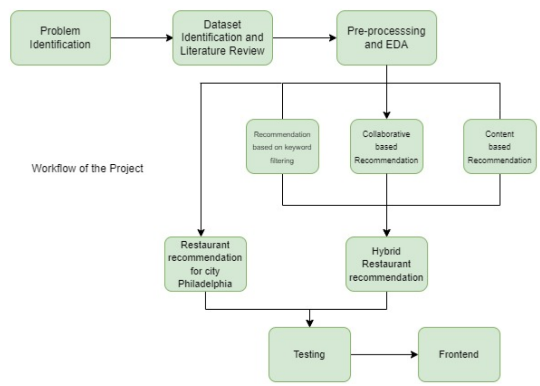


Figure 11: Workflow of the Project

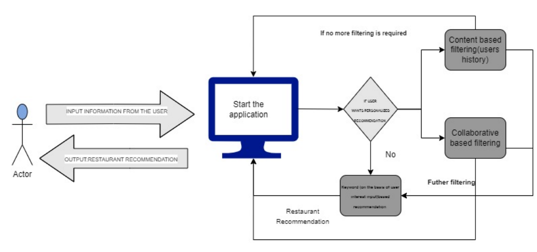


Figure 12: Flow chart from user's point of view

6 Code

Following are some snippets from code of hybrid recommendation:

```python
def recommender_interface():
    """
    User interface for the hybrid recommendation engine.
    """

    # initiation
    print("Hey, welcome to the Hybrid Yelp Recommender!")
    boolean = True
    personalized = False
    original_score = False
    n = 5
    # initiate a recommender object
    print("Please wait while we initiate the recommendation engine\n loading...")
    recommender = Recommender(personalized=True)
    print("Yeah, recommendation engine is ready to use!")

    # def nested function for obtaining user keywords for filtering the recommendations
    def _getkeywords():
        # initiate
        zipcode, city, state = None, None, None
        max_distance = 10
```

Figure 13:

```python
    # main user interface for the recommendation engine
    while boolean:

        r = input("Want to try a customized recommendation based on your Yelp user history? yes/no\n")
        if r.startswith('Y') or r.startswith('y'):
            personalized = True

        # personalized recommender modules
        if personalized:
            print("Awesome! Let's start your personalized recommendation.")
            # obtain user id
            r = input("To retrieve your user history, please enter your Yelp User ID (length of 22 characters):\n")
            if len(r) == 0:
                print("Ooops, no user id is provided! Let's give it another try.")
                continue
            elif len(r) != 22:
                print("Ooops, it seems to be an invalid user id! Let's give it another try.")
                continue
            else:
```

Figure 14:

```python
    # non-personalized recommender module
    else:
        print("That's cool! Let's filter by keywords and generate your recommendations!")
        zipcode, city, state, max_distance, cuisine, style, price = _getkeywords()
        print("Great! Filtering criteria fetched! Just one more question before generating your recommendations")
        r = input("Wanna rank your recommendations by 'smart' ratings?\n'smart' rating adjusts the original restaurnat average star rating by
        if r.startswith('N') or r.startswith('n'):
            original_score = True
        print("Awesome, all set! Here is your recommendations:\n")
        print("---------")
        result = recommender.keyword(zipcode=zipcode, city=city, state=state, max_distance=max_distance, cuisine=cuisine, style=style, price=
        print("---------")

    # refine recommendation results
    if result is not None and len(result) > 0:
        r = input("Would you like to display more/less recommendation results? Enter the desire number to continue or any other key to skip:\
        try:
            n = int(r)
            print("---------")
```

Figure 15:

7 Evaluation Metrics

The following performance metrics were used in this project:

7.1 For Non-Personalization(keyword-based filtering)

Test results demonstrate that the recommended restaurants are the only ones that match the user's keyword combination and are ranked according to the appropriate adjusted scores.

7.2 Personalized Recommendation

For personalized recommendation, we use following metrics:

- RMSE:We used the RMSE as an evaluation metric to find the accuracy of ratings. RMSE is a measure of the difference between the predicted ratings and the actual ratings of the restaurants. It is a widely used metric in recommendation systems as it gives a measure of the accuracy of the predicted ratings.
  The formula for RMSE is given as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (\text{ Predicted }_i - \text{ Actual }_i)^2}{N}}$$

- NDCG(Normalized Discounted Cumulative gain):nDCG is a widely used metric in recommendation systems that takes into account the relevance and ranking of the recommended items. It measures the quality of the ranking of recommended items, where a higher score indicates better performance. The formula for nDCG is given as:nDCG@k = DCG@k / IDCG@k Where k: The number of recommended items to consider DCG@k: Discounted Cumulative Gain at k, which is the sum of the relevance scores of the top k recommended items IDCG@k: Ideal Discounted Cumulative Gain at k, which is the DCG@k when the recommended items are in perfect order based on relevance The model used to predict ratings for test sets with new users and businesses as well as testsets without new users or businesses. Following the computation of NDCG@top10 and NDCG@5, the recommendation ranking is constructed for each user in the testset based on the projected ratings in decreasing order.

7.3 Collaborative Filtering

Rmse achieved by models after baseline results:

- Simple SVD: -2.0039
- SVD with bias -1.2305
- Grid search (SVD):1.23
- NMF without bias -1.41
- NMF with bias-1.55
- NMF using scikit learn-2.79
- NMF model(grid search optimization):1.4037

Average nDCG achieved by our best model at nDCG@5 AND nDCG@10 is 0.96 and 0.94 respectively.

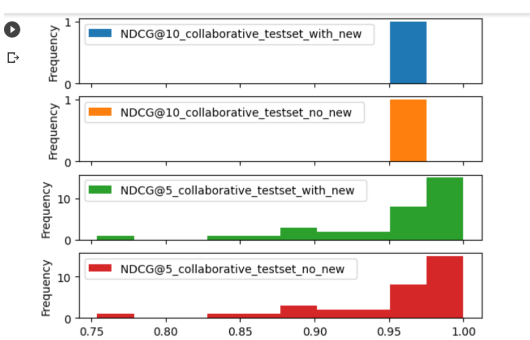


Figure 16: NDCG score for colaborative filtering over the test set

7.4 Content Based Filtering

Average nDCG achieved by our best model at nDCG@5 AND nDCG@10 is 0.86 and 0.85 respectively.(Figure 17)

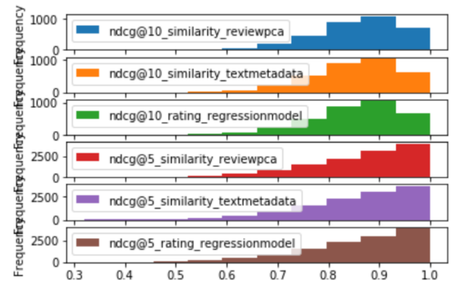
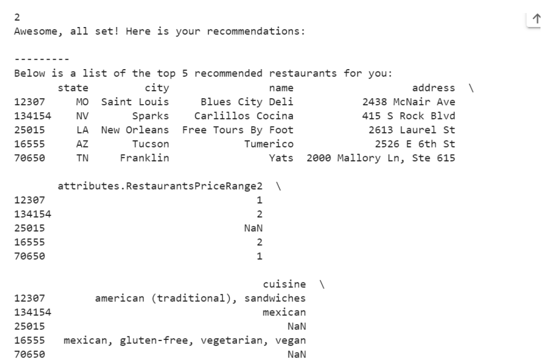


Figure 17: NDCG score for content filtering over the test set

8 Results

Refer Figure 18.



```
2
Awesome, all set! Here is your recommendations:

---------
Below is a list of the top 5 recommended restaurants for you:
       state         city                 name                  address  \
12307     MO  Saint Louis      Blues City Deli           2438 McNair Ave
134154    NV       Sparks     Carlillos Cocina           415 S Rock Blvd
25015     LA  New Orleans  Free Tours By Foot            2613 Laurel St
16555     AZ       Tucson             Tumerico            2526 E 6th St
70650     TN     Franklin                 Yats  2000 Mallory Ln, Ste 615

       attributes.RestaurantsPriceRange2  \
12307                                  1
134154                                 2
25015                                NaN
16555                                  2
70650                                  1

                                      cuisine  \
12307       american (traditional), sandwiches
134154                                 mexican
25015                                      NaN
16555   mexican, gluten-free, vegetarian, vegan
70650                                      NaN
```

Figure 18: Output of Restaurant Recommendation =

9 Testing of the Application

The personalized recommendation often takes 2 to 3 seconds to return, whereas the non-personalized recommendation just takes one second. A variety of restaurants are listed in the results of the recommendation. The list of suggestions for users with very little preference history resembles the general list for new users.

10 Conclusion

A user-friendly interface is developed to integrate the three filtering techniques, gather user interests, and guide users through the hybrid recommendation engine using user-interactive questions.
The following aspects of this recommendation system are available to new or anonymous users: the recommendation engine can offer base-case recommendations utilising restaurant attributes and/or location information. Users can choose whether to utilise the collaborative personalization or the content-based personalization to deliver personalised suggestions when providing their user ID as input and their rating history from the database. Smart weighted ratings will be determined by combining the quantity of ratings ('popularity') with the average rating ('quality').

11 Future Work

- Personalization based on tips and check ins can also be integrated.
- More recent reviews are seen as more reliable than earlier reviews since factors like restaurant ownership, food quality, service, and the environment can all change over time. Therefore, by including the age relevance of the rating, the rating measure may be further enhanced.

References

[1] Ahsan Habib, Abdur Rakib, and Muhammad Abul Hasan, "Location, Time, and Preference Aware Restaurant Recommendation Method", IEEE, 19th International Conference on Computer and Information Technology, pp. 315-319, 2016.

[2] Nanthaphat Koetphrom, Panachai Charusangvittaya, Daricha Sutivong, "Comparing Filtering Techniques in Restaurant Recommendation System", IEEE, pp. 46-51, 2018

[3] Khushbu Jalan, Kiran Gawande, "Context-Aware Hotel Recommendation System based on Hybrid Approach to Mitigate ColdStart-Problem", IEEE, Communication, Data Analytics and Soft Computing, pp. 2364-2369, 2017.

[4] Jun Zeng, Feng Li, Haiyang Liu, Junhao Wen, Sachio Hirokawa, "A Restaurant Recommender System Based on User Preference and Location in Mobile Environment", 5th IIAI International Congress on Advanced Applied Informatics, IEEE, pp. 55-60, 2016

[5] Ling Li, Ya, Zhou, Han Xiong, Cailin Hu, Xiafei Wei, "Collaborative Filtering based on User Attributes and User Ratings for Restaurant Recommendation", IEEE, pp. 2592-2596, 2017.

[6] A. Poriya, T. Bhagat, N. Patel, and R. Sharma, 'Non-personalized recommender systems and user-based collaborative recommender systems', Int. J. Appl. Inf. Syst, vol. 6, no. 9, pp. 22–27, 2014.

[7] A. Pavate, A. Chaudhary, P. Nerurkar, P. Mishra, and M. Shah, 'Cuisine Recommendation, Classification and Review Analysis using Supervised Learning', in 2020 International Conference on Convergence to Digital World-Quo Vadis (ICCDW), IEEE, 2020, pp. 1–6.

[8] A. Gupta and K. Singh, 'Location based personalized restaurant recommendation system for mobile environments', in 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, pp. 507–511. doi: 10.1109/ICACCI.2013.6637223.