UNKI

Ecole Polytechnique Fédérale de Lausanne

EPFL

Unki

A restaurant recommender system for individuals, groups and diversity

by

Student Name	Student Sciper
Alexandre Reis de Matos	282552

Supervisor: Anne-Marie Kermarrec
Project Duration: September, 2022 - January, 2023
Faculty: School of Computer and Communication Sciences, EPFL

Cover : Unki logo (Modified)
Style: EPFL Report Style, with modifications by Batuhan Faik Derinbay

Abstract

In this work, a recommendation system for restaurants was developed. Multiple approaches were explored, including user-based collaborative filtering, item-based collaborative filtering, and diversity-based recommendation. The performance of the system was evaluated using a custom metric that takes into account the subjective nature of food preferences and the overall dining experience. The results show that the system was able to provide accurate and diverse recommendations for individual users and groups. In order to further improve the system, online testing and the incorporation of additional data sources could be explored.

Contents

1
Introduction

Recommendation systems have become an essential tool for providing personalized recommendations to users in various industries. The goal of this project is to construct a restaurant recommendation system using data scraped from the web as part of a comprehensive effort to identify the most effective approaches for developing recommendation systems for the restaurant industry.

1.1. Mobile application

The restaurant recommendation system aims to provide personalized recommendations to users in the form of a mobile application. This application will allow users to quickly and easily receive recommendations on where to dine and drink, taking into consideration their past consumption history. This approach differs from current solutions, such as TheFork, in that it enables users to actively request recommendations based on specific criteria rather than simply being presented with recommendations while using a website or app. By streamlining the process of obtaining a recommendation down to a single click and transitioning from a passive to an active recommendation model, this system has the potential to enhance the user experience and increase customer satisfaction.

1.2. Functionalities

The mobile application offers three primary categories of recommendations: individual, group, and diversity. Individual recommendations are tailored to provide a suitable restaurant recommendation for a single user or a group of individuals who trust the user's judgment. Group recommendations consider the preferences and needs of the entire group and can be customized with biases to prioritize certain members' interests. Diversity recommendations, the most challenging category, aim to suggest restaurants that are novel to the user but still align with their preferences. The level of diversity in these recommendations can be adjusted to each user's desired level of experimentation.

Overall, the purpose of these recommendations is to provide users with efficient access to personalized suggestions based on their particular needs and desires.

2

Dataset extraction

The accuracy and effectiveness of a recommendation system heavily depends on the quality of the data used to train and evaluate the system. In the context of restaurant recommendation systems, data on user preferences and past orders is essential for generating personalized recommendations. However, obtaining high-quality data can be a challenging task. Most of the companies contacted in the scope of this project were unwilling to share their data, due to concerns about privacy and competitive advantage. As a result, the choice was made to turn to alternative sources of data, namely web scrapping. While this can be a useful solution in some cases, it is important to note that data from the web may not always be of the best quality. It may be noisy due to fake reviews or incomplete, which can affect the accuracy and effectiveness of the recommendation system.

2.1. Existing datasets

After conducting thorough research, it was determined that no readily available datasets on the internet were suitable for this specific project. A dataset containing data from Bangalore [10] was identified, but it did not contain any user data that would have allowed for the creation of users' histories of consumption. Additionally, the dataset was limited to Bangalore, while the case study for this project is focused on Switzerland and specifically Lausanne. Another dataset provided by Yelp [13] was also reviewed, but it was limited to regions that were not relevant to this project, including metropolitan areas centered on Montreal, Calgary, Toronto, Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison, and Cleveland. As a result, the decision was made to proceed with web scraping as the primary method of data collection.

2.2. Data source

In order to determine the most suitable website for sourcing data, various options such as TripAdvisor, Google, Yelp, and FourSquare were reviewed and compared. Ultimately, it was determined that TripAdvisor offered the most detailed information on each restaurant and the most publicly available user data. Therefore, the focus was placed on utilizing TripAdvisor as the primary source for data collection.

2.3. TripdAdvisor's data structure

TripAdvisor organizes restaurants according to their country and city, with lists of restaurants available for each city and lists of all cities available for each country. Upon accessing the page for a specific restaurant, it is possible to find all or at least some of the information listed in table 2.1.

Restaurant page	Review	User profile
1. Restaurant name	1. Real name	1. Real name
2. Number of reviews	2. Rating	2. Username
3. Address	3. Title of review	3. Age category
4. Price range	4. Text content of review	4. Sex
5. Contact information	5. Pictures	5. Region of origin
6. Website	6. Number of reviews	6. Account creation date
7. Menu	7. Date of the visit	7. User level
8. Cuisine types	8. Date of the publication	8. Total number of reviews
9. Opening times	9. Device used	9. Number of cities visited
10. Meal types		10. Upvotes
11. Functionalities		11. Number of pictures
12. Overall rating		12. Categories of the user
13. Cuisine rating		
14. Service rating		
15. Quality / Price ratio		
16. Mood rating		
17. Michelin or not		
18. Special regimes		

Table 2.1: Description of data available on TripAdvisor.

A few notes on the elements of Table 2.1 and their meanings:

- **Meal types:** breakfast, lunch, or dinner
- **Functionalities:** take-out, accepting credit cards, accessibility for individuals with disabilities, etc.
- **Special regimes:** vegetarian-friendly, vegan-friendly, etc.
- **Number of reviews:** the number of reviews posted by the user who wrote the review being analyzed
- **User level:** based on user activity
- **Upvotes:** the number of people who found the user's reviews to be useful
- **Number of pictures:** the number of pictures of dishes uploaded by the user in reviews
- **Categories:** e.g. "friend of nature," "family holidays," etc.

Criteria 12 through 16 are defined by the ratings given by users in their reviews. When clicking on the user's name or picture in a review, an additional box may be displayed with further information on the user who published the review. However, it should be noted that these details do not appear for all users, and each restaurant may have anywhere from zero to hundreds of pages of reviews.

2.4. Data collection

For this project, not all publicly available attributes were utilized. Data was collected and stored in three separate CSV files, as shown in Table 2.2.

A few more notes on the attributes of table 2.2 and their meanings:

- **Restaurant url:** since it is unique, it is used as the ID
- **Username:** since it is also unique, it is used as the ID

In addition to these user attributes, two others were added following careful analysis of users' histories of consumption:

- **Preferred cuisines:** among the restaurants the user has visited, how were the different types of cuisines distributed
- **Average ratings:** over the restaurants the user has visited, for overall ratings, cuisine ratings, service ratings, quality/price ratio, mood ratings, price range, and the presence of a Michelin star

For every attribute in all three CSV files, if the information was missing, the default value used was "None" or 0 for numerical attributes.

restaurants.csv	reviews.csv	users.csv
1. Restaurant name	1. Restaurant name	1. Name
2. Restaurant url	2. Restaurant url	2. Username
3. Address	3. Rating given in the review	3. Registration date
4. Price range	4. Review date	4. Number of reviews
5. Overall rating	5. Visit date	5. Number of cities visited
6. Number of reviews	6. Title	6. Categories
7. Price range	7. Text content of review	7. Country
8. Cuisine types	8. Link to uploaded images	8. Canton
9. Functionalities		9. City
10. Special regimes		10. Age
11. Meal types		11. Sex
12. Cuisine rating		
13. Service rating		
14. Quality / Price ratio		
15. Mood rating		
16. Michelin or not		

Table 2.2: Description of the dataset created from the TripAdvisor data

2.5. Web scrapping

The next step in the process was to implement the technical aspect of data collection. The first concern was to assess the level of protection in place on TripAdvisor and determine the measures that would need to be taken in order to successfully scrape data from the website. It was determined that TripAdvisor only required the presence of a user agent in the request's header, which meant that there was no need to use IP rotation techniques, user agent rotation techniques, or randomization of the time interval between requests. Additionally, there was no limit on the number of requests per day. The primary bottleneck of the process was at the page loading stage, as there was some delay until the pages were fully loaded, and some elements had to be clicked in order to be displayed. The web scraper was developed in Python using the Selenium library [9] for requests and the BeautifulSoup library [11] for data parsing. The process was completed in three steps using three custom-made web scrapers:

1. **Cities:** Obtain a list of all Swiss cities on TripAdvisor, along with the link to the page with all the restaurants in those cities
2. **Restaurants:** From the page of each city, obtain the links to every restaurant page and store some basic information on the restaurants
3. **Data extraction:** From each restaurant page, extract all the attributes mentioned in section 2.4

The first part took a few days to implement and run. The second part took approximately one week to implement and run. The final web scraper was more complex and took approximately four weeks to build and run. All the web scrapers ran continuously, but were occasionally interrupted by bad requests or code errors, which were resolved within 4-5 hours.

2.6. Extracted dataset characteristics

TripAdvisor is known for the potential bias of fake reviews on restaurants' ratings. However, it is estimated that these fake reviews represent less than 10% of all reviews, according to TripAdvisor's 2021 transparency report [3]. As a result, the potential bias from fake reviews was not considered at this early stage.

Although a restaurant, review, or user may possess numerous attributes, it is common for them to only have a limited number of them. As shown in Figure 2.1, barely half of the users publish their city of origin, and only 20% make their age and gender publicly available. As a result, it was decided that gender and age would not be used until a more complete dataset was available. The city and country of origin could be used to filter recommendations upon request, but no filter was applied in the results presented in this report.

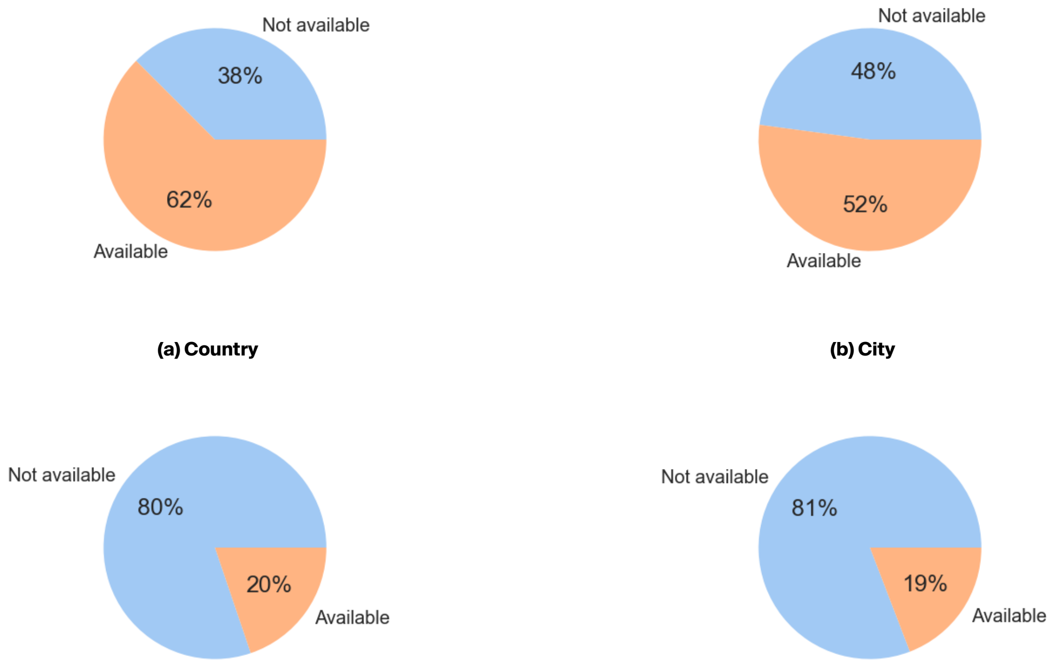
Similarly, for restaurants, we can observe phenomena akin to those previously mentioned in Figure 2.2 for some ratings. Cuisine types are present 90% of the time. Interestingly, we find similar statistics for cuisine rating, service rating, and quality/price ratio, slightly above 60%. This similarity may be attributed to the fact that these ratings are an aggregation of users' individual ratings, and that a user who takes the time to rate the cuisine will often also rate the service and the quality/price ratio.

On the other hand, mood is only rated 25% of the time. This low frequency may be due to the subjectivity of this metric, which may make it difficult to accurately assess. As a result, it was decided not to use mood as an attribute for the work presented in this paper.

Finally, the Michelin parameter is the least available attribute. It was assumed that if a restaurant had at least one Michelin star, it would be mentioned on TripAdvisor. Therefore, any restaurant with no information on Michelin stars likely does not have any, making the Michelin parameter still useful for the remainder of the project.

2.7. Dataset in numbers

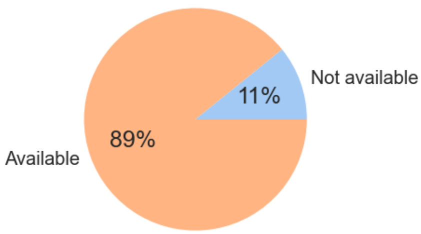
The final dataset consisted of 31'643 restaurants, 591'520 reviews, and 349'995 users. Out of all the users in the dataset, approximately 5'000 have written a sufficient number of reviews (10 or more) to be considered active reviewers. These active reviewers have contributed a total of 84'000 reviews.



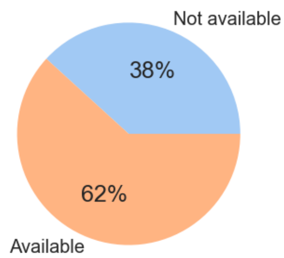
(a) Country (b) City

(c) Age (d) Sex

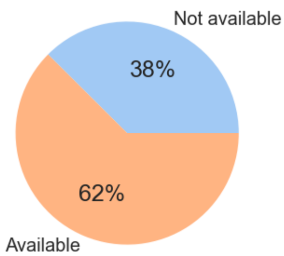
Figure 2.1: Statistics for user key attributes. Every pie chart represents the number of users with the attribute information publicly available vs the number of users without the attribute information publicly available



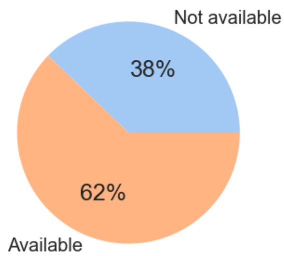
(a) Cuisine type



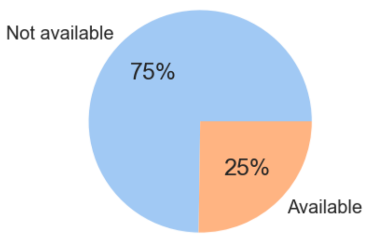
(b) Cuisine rating



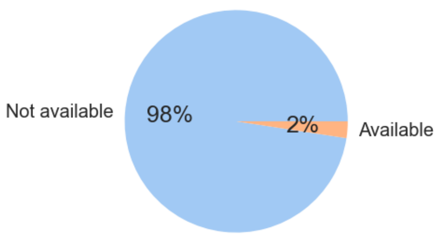
(c) Service rating



(d) Quality/Price ratio



(e) Mood rating



(f) Michelin rating

Figure 2.2: Statistics for restaurant key attributes. Every pie chart represents the number of restaurants with the attribute information publicly available vs the number of restaurants without the attribute information publicly available

3
Evaluation metric

Evaluating the performance of a recommendation system is a crucial task, as it enhables an acurate assement of the accuracy and effectiveness of the system. Classic metrics such as precision, recall, or hit rate may not always be sufficient or relevant for accurately evaluating the performance of a restaurant recommendation system. For example, they may not take into account the subjective nature of food preferences or may not adequately capture the user's overall dining experience. To address these limitations, a custom metric was developed.

3.1. Offline evaluation

Offline evaluation can be a useful method for evaluating the performance of a recommender system, particularly when it is not possible to conduct online testing. However, it is important to keep in mind the limitations of this approach. One key limitation is that offline evaluation may not accurately reflect the performance of the system in a real-world setting, as the ground truth data used may not fully capture the complexity and variability of user behavior. Additionally, offline evaluation does not provide the opportunity to continually improve the system through feedback from actual users. In this case, at the time of evaluation, it was not possible to have a platform for conducting online testing. However, in the absence of such a platform, offline evaluation can still provide valuable insights into the performance of the recommender system and can be a useful tool for identifying areas for improvement.

3.2. User vector

Each user has visited at least one restaurant, meaning it is possible to describe a user's preferences by examining the restaurants they have visited. To do so, a vector was created for each user based on their consumption history. The higher the rating given by the user to the restaurant, the higher the weight of the restaurant in the user vector. The components of the vector are described in figure 3.1.

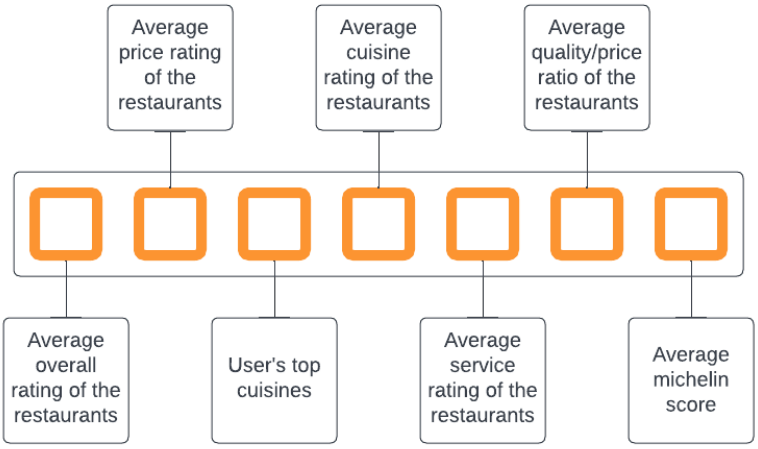


Figure 3.1: User vector

Each restaurant was assigned a value of 1 if it is a Michelin restaurant and a value of 0 if it is not. The Michelin score for a given user was calculated as the average of these values for all the restaurants the user had visited.

3.3. User's top cuisines

To determine the user's preferred cuisines, a distribution was created based on the popularity of different cuisines among the restaurants the user had visited. This distribution was specific to each user, so not all users had all 149 available cuisine types in their distribution. In order to measure the distance between the cuisine of a recommendation and the user's preferred cuisine distribution, the following process was followed:

```
cuisine = cuisine of the recommendation
distribution = distribution of user cuisines

if cuisine is in distribution :
    return 1 - distribution(cuisine)
else :
    return 1 # Maximal distance
```

Therefore, the higher the value in the user's cuisine distribution, the smaller the distance between the recommendation and the user's preferred cuisines. If the cuisine is not present in the user's distribution, the recommendation receives the maximum penalty for being completely off in terms of cuisine preference.

3.4. Error penalization

Errors in the recommendations are heavily penalized in order to obtain a meaningful metric for evaluating the accuracy of each parameter i for user u. The following formula is used for this purpose:

$$error_{u,i} = recommendation_{u,i} - vector_{u,i}$$

$$accuracy_{u,i} = \frac{3}{5} * exp^{exp(abs(error_{u,i}))} - \frac{3}{5} * exp(1)$$

where $recommendation_{u,i}$ is the i^{th} component of the recommendation vector for user u and $vector_{u,i}$ is the i^{th} component of the user vector for user u. The coefficients and formula were chosen through trial and error to ensure that the accuracy of random recommendations is on average below 50%.

3.5. Error weights

To account for the differing biases towards different parameters, each user is associated with a variance vector. This vector represents the variability of a given parameter among the restaurants visited by the user. For example, if a user frequently visits restaurants with a wide range of prices, the variance of the "price range" parameter for that user would be high. On the other hand, if the user only visits restaurants within a narrow price range, the variance of the "price range" parameter would be low.

For each component of the user vector, the value of the component in the restaurant vectors is examined for each restaurant the user has visited. If the variance of a given parameter is large, it indicates that the parameter is of low importance to the user, as the user has visited restaurants with a wide range of values for that parameter. In this case, the parameter is given a lower weight when calculating the distance between the recommendation and the user. On the other hand, if the variance is small, it indicates that the parameter is of high importance to the user and is given a higher weight. The weights are defined as follows for each parameter i of user u :

$$w_{u,i} = \frac{1}{variance_u(i)}$$

3.6. Custom metric summary

The user vector represents the characteristics and preferences of a given user, as determined by the system based on the user's history of consumption. The weight vector is used to reflect the relative importance of different parameters to the user. It is based on the variance of the same history of consumption that was used to define the user vector. The final accuracy result for each user u is calculated by taking into account both the user vector and the weight vector, through the use of a weighted distance formula:

$$accuracy_u = \frac{1}{\sum_i w_{u,i}} * \sum_i w_{u,i} * accuracy_{u,i}$$

This allows the system to tailor its recommendations to the specific biases and preferences of each individual user.

4

Individual recommendation

Recommendation systems have become a crucial tool in the field of information retrieval, with the goal of providing personalized recommendations to users based on their preferences and past behavior. In the context of the restaurant industry, individual recommendation systems can be used to suggest dishes or restaurants to customers based on their past orders and ratings, with the aim of enhancing the customer experience and increasing customer satisfaction.

4.1. Background and related Work

Previous research ([2], [4], [1]) has been conducted in this area using various algorithms such as support vector machines, mainly based on overall ratings of restaurants without taking much into account other parameters such as cuisine type, price range, or whether the restaurant has a Michelin star. This has motivated this project to pursue further research and custom recommender setups while still leveraging known algorithms.

4.2. Baselines

To establish a benchmark for evaluating the performance of the recommendation system, two control algorithms were implemented: a random recommender and a popularity-based recommender. The random recommender selects a restaurant from the available options randomly, while the popularity-based recommender generates a probability distribution based on the number of reviews received by each restaurant and recommends a restaurant with a probability proportional to its popularity. These control algorithms provide a reference point for assessing the performance of the recommendation system and ensure its proper functioning.

4.3. User-based recommendation

To implement the first recommendation system, a user-based collaborative filtering algorithm was chosen and it was based on a weighted k-Nearest Neighbors algorithm with $k = 30$. This value of k was determined through cross-validation with various parameter values. The data was represented as a sparse matrix, with ratings given by a user to an item at the intersection of the user row and item column, and zeros elsewhere. Cosine similarity was chosen as the distance metric, as it considers users with similar tastes but varying levels of expectation as being similar. The algorithm then generates a ranking of the most popular restaurants among the nearest neighbors in decreasing order, based on the frequency of each restaurant and the distance of the neighbors to the user, which serve as weights. The top restaurant in the ranking is recommended to the user.

4.4. User-restaurant space

To enhance the capabilities of both user-based and item-based recommendations, the previously created user vectors were leveraged to construct a space where both restaurant vectors and user vectors could coexist. This significantly improved the performance of the recommendation system.

4.5. Item-based recommendation

To leverage the user-restaurant space, the user vector was treated as an item vector and a modified version of the user-based recommendation algorithm was employed. The parameters of the algorithm remained unchanged, except for the distance metric, which was altered to Euclidean distance. This was done in order to shift the focus from taste to the cumulative distances of each parameter. The modified algorithm proceeded as follows: first, the k-nearest neighbors to the user vector among the restaurant vectors were identified using Euclidean distance. Next, new values of the neighbors were calculated using the custom evaluation metric. Finally, the list of neighbors was sorted according to these new values and the top of the sorted list was recommended to the user. It is important to note that the cuisine parameter was excluded from the neighbor search stage due to the difficulties in comparing it. The additional step of sorting the recommendations based on the custom metric was necessary to ensure the accuracy of the recommendations, as the custom metric takes user weights into account in addition to the Euclidean distance metric used by the k-nearest neighbors algorithm.

4.6. User-item mixed recommendation

The final recommendation system implemented was a hybrid model that also leveraged the user-restaurant space. Rather than finding the k-nearest users or items, this model searches for the k-nearest vectors, which could be either users or items. The process for making recommendations is similar to that described in section 4.5. If the top of the final list (after sorting according to the custom metric) is a restaurant, it is recommended to the user. If it is a user, the algorithm examines the neighbor's list of restaurants and recommends the one with the most similar vector to the user vector based on the custom metric. As with the item-based recommendation, the cuisine parameter was excluded for comparability reasons.

4.7. Results and analysis

As shown in Figure 4.1, the item-based and user-item mix algorithms outperform the other algorithms, with the exception of the Michelin parameter. This is because only 2% of the restaurants in the dataset have a Michelin star, as depicted in Figure 2.2f. Therefore, an algorithm that always recommends a restaurant with a Michelin score of 0 will be correct 98% of the time.

The user-based approach performs comparably to the popularity baseline and significantly worse than the item-based approach. This may be due to the following factors: the use of sparse, high-dimensional data vectors compared to the low-dimensional vectors in the user-restaurant space, the distance metric (cosine similarity for the user-based approach and Euclidean distance for the others), and the sorting of the recommendations based on popularility rather than the custom metric.

Overall, the item-based recommendation system slightly outperforms the user-item mix recommendation by approximately 1%. The item-based system performs well for all parameters except cuisine type, for which the user-item mix recommendation outperforms it by nearly 20%. Although the item-based recommendation system may be beneficial for many consumers, it does not perform well for those who prioritize cuisine type. As a result, the mix-based recommendation was selected for the remainder of the project due to its consistent performance across all parameters.

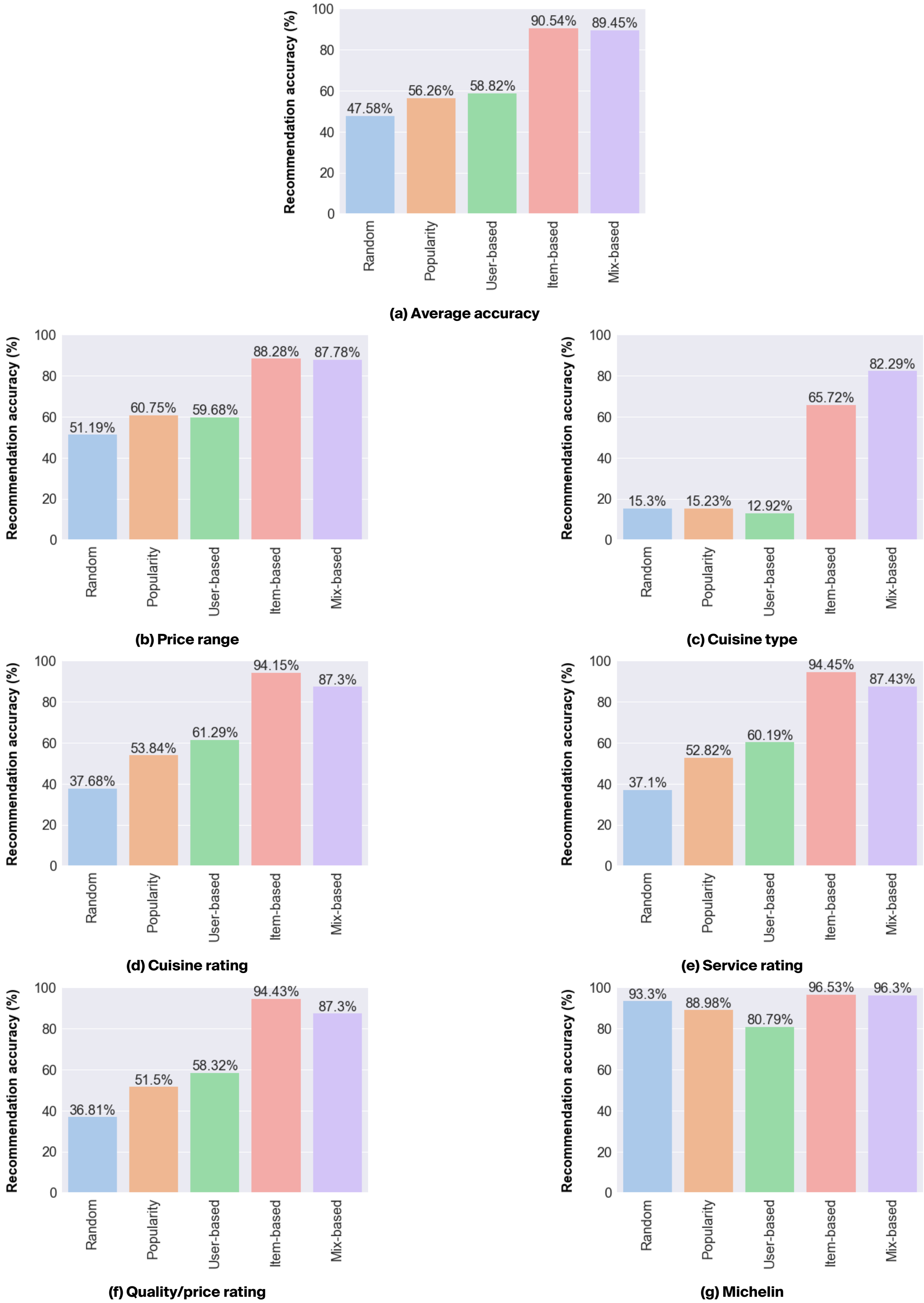
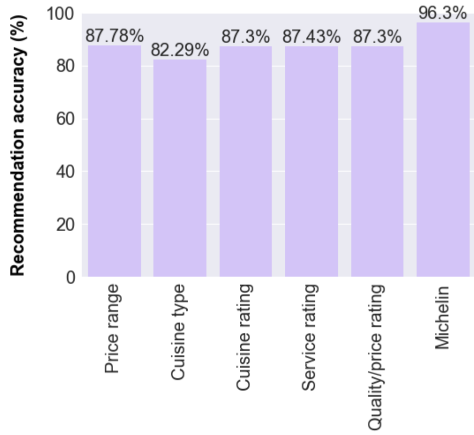
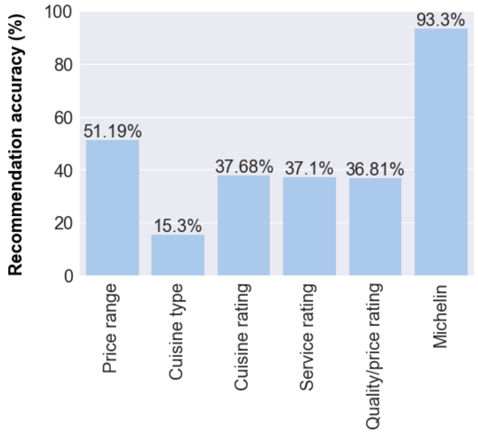


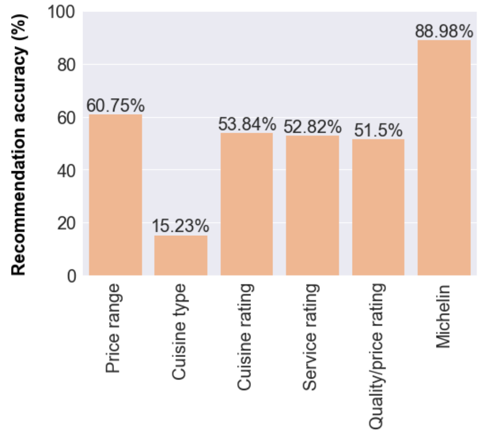
Figure 4.1: The average accuracy achieved for all algorithms, as well as the accuracy achieved per individual parameter



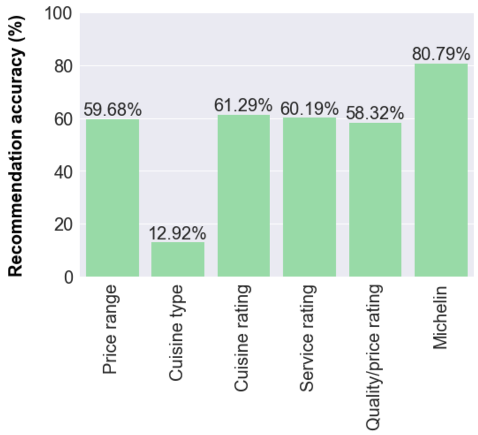
(a) Mix-based recommender



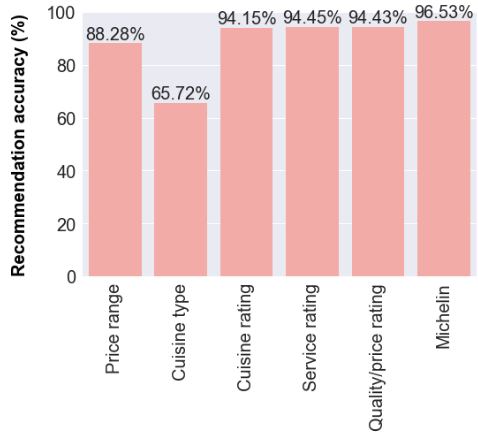
(b) Random recommender



(c) Popularity-based recommender



(d) User-based recommender



(e) Item-based recommender

Figure 4.2: Another view of the accuracy of the algorithms : the average accuracy per parameter achieved for every algorithm

5
Group recommendation

Group recommendation systems aim to provide personalized recommendations to groups of users by taking into account the preferences of all group members. These systems have the potential to improve the collaborative experience of users and increase the diversity of recommendations. There are several reasons why group recommendation systems have not yet become widespread. One reason is that developing effective group recommendation algorithms is a challenging task, as it requires modeling the complex interactions between group members and their preferences. This can be difficult to do accurately, especially when the group is large or the preferences of the group members are highly diverse. Despite this, there have been recent developments that may shift the status quo. In 2022, Spotify introduced a new feature that allows users to generate algorithmic playlists for sharing among groups of friends. These playlists are designed to cater to the preferences of all members within the group. This innovation may inspire other major companies specializing in content recommendation to consider implementing similar features.

5.1. Background and related work

In the realm of group recommendation, the memory-based method offers two approaches [12]: preference aggregation and score aggregation. Research on the score aggregation method has sought to automatically identify communities and make recommendations to these communities as groups based on each user's past interactions (e.g. [7], [6]). These recommendations are made by predicting the scores that both individual users and the group as a whole would give to items. In contrast, preference-based approaches such as CoFeel ([14]) focus on forming groups based on emotions and personal preferences.

5.2. Challenges

As previously noted by Hallström ([4]), the problem of group recommendation is highly multifaceted and can incorporate a wide range of psychological and philosophical considerations. There is no one-size-fits-all solution to group recommendation, as it largely depends on the nature of the event (e.g. birthday celebration, romantic dinner, child-friendly outing), the individuals involved, and their relationships. In this paper, two methods are explored, but their true efficacy can only be determined through online testing.

5.3. Group recommendation algorithm

This study examines memory-based group recommendation through preference aggregation. Two approaches are compared. The first approach follows the method proposed by Ludovico Boratto ([5]), which involves obtaining a list of recommendations for each member of the group and then selecting the group recommendation based on the criteria outlined in Section 5.4. The second approach involves creating a single group vector by taking the average of all individual user vectors, and then making recommendations for the group using this vector in the same manner as recommendations are made for individual users. Random sampling is used to select groups of users with at least 10 reviews, and group sizes ranging from 2 to 5 people are evaluated. The algorithm used to generate recommendations is the one described in Section 4.6. The effectiveness of group recommendations is quantitatively evaluated as the average success rate of recommendations for each member of the group.

5.4. Evaluation criteria

Given the aggregated list of recommendations for all users in the group, multiple criteria are calculated for the users in the group based on the restaurants from the aggregated list. These criteria are based on the recommendations of Hallström ([4]) and are as follows:

1. **Average satisfaction:** Recommend the restaurant that maximizes the unweighted average accuracy over the group
2. **Least misery:** Recommend the restaurant that maximizes the lowest accuracy value for a member of the group, setting a threshold on the minimum accepted accuracy for the group
3. **Maximum pleasure:** Recommend the restaurant that maximizes the highest accuracy value for a member of the group, allowing for large discrepancies in accuracy between group members
4. **Average disagreement:** Minimize the pairwise difference in accuracy between users

5.5. Baseline

Although personalized recommendation is expected to perform at least as well as the result obtained in Section 4, this is not the case in the current study. Personalized recommendation actually performs 4% better than the result in Section 4. As mentioned in Subsection 4.6, the k-Nearest Neighbors algorithm uses Euclidean distance as the metric for finding the nearest neighbors. However, the accuracy of the recommendations is evaluated using a different metric that takes into account user weights. These weights account for the discrepancy between the two metrics. In a group setting, there may be recommendations from other users that perform better on the final metric with weights, leading to the outperformance of personalized recommendation at this stage. Despite this, personalized recommendation still performs better than any of the group recommendations and is stable, so it serves as the baseline for comparison with group recommendations.

5.6. Results and analysis

The performance of the algorithm was evaluated using different criteria on 100 different groups for each group size, with a limit of 30 recommendations per user. These limits were imposed due to performance constraints, and are considered to be relevant since the baseline remains stable around 94% accuracy. Additionally, the results were not improved by additional iterations of the experiment.

As shown in Figure 5.1, the average accuracy of group recommendations does not significantly decrease with increasing group size. The criteria of least misery, maximum pleasure, and average satisfaction perform better than personalized recommendation in chapter 4 for the same reaons discussed in Section 5.5. However, it is important to note that the accuracy based on these criteria should be interpreted with caution, as the results are highly dependent on the specific occasion and latent parameters not known to the recommendation system. It is not possible to definitively state that one option is superior to the others.

For example, using least misery or maximum pleasure as the decision-making criteria can still result in significant differences between individual users within the group. For instance, in a group of 5 people, if the lowest accuracy is 60% and all others have an accuracy of 98%, the average accuracy of the group is still above 90%, which is a good result at the group level, but not necessarily at the individual level. This highlights the content-dependent nature of these criteria. The same concept applies to maximum pleasure and average satisfaction.

Finally, average disagreement performs similarly to the custom metric and minimizes the pairwise difference in accuracy between group members. Additionally, the custom metric is faster to compute, making it a good alternative if average satisfaction is the desired criterion. However, there is a trade-off between performance and reducing the gap in user satisfaction, as the performance is up to 10% worse. If the group's needs are prioritized over individual preferences, this criterion may be the most suitable option.

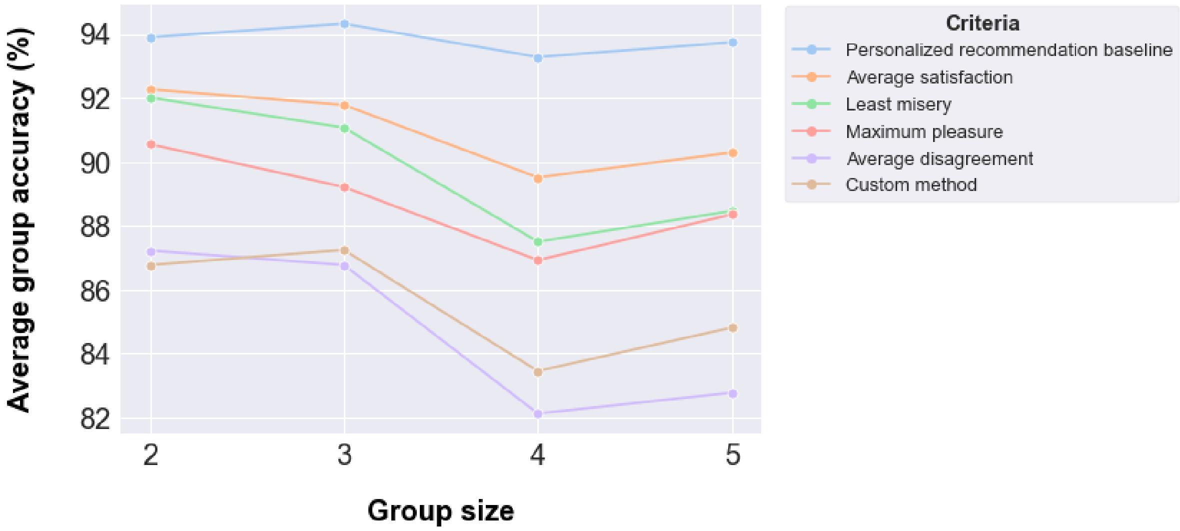


Figure 5.1: Average accuracy of the recommendations for groups based on different criteria

5.7. Next steps

Instead of modifying the individual recommendation algorithm to use the custom metric to measure distance from the beginning, it could be interesting to explore the possibility to adopt a group strategy to improve the effectiveness of personalized recommendation.

6

Diversity recommendation

Diversity in recommender systems refers to the inclusion of a wide range of items or content in the recommendations provided by the system. This is important because it can help to reduce the risk of homogeneity in the recommendations, which can lead to a less interesting and engaging user experience. A diverse set of recommendations can also help to expose users to new and unexpected items, which can be a valuable source of discovery and learning. In addition to benefiting users, diversity in recommender systems can also be beneficial for businesses, as it can help to increase the overall diversity of the products that are being consumed, leading to a more diverse and sustainable ecosystem.

6.1. Background and challenges

There are a few reasons why diversity may not be a prominent feature in many recommender systems today. One reason is that optimizing for diversity can sometimes be at odds with optimizing for relevance. Recommender systems are often designed to provide users with recommendations that are as relevant as possible to their interests and preferences. This can sometimes mean that the recommendations are highly similar to one another, as the system is trying to surface items that are closely related to what the user has liked or interacted with in the past. Incorporating diversity into the recommendation process can require trade-offs, as the system may need to consider a wider range of items and potentially sacrifice some level of relevance in order to provide a more diverse set of recommendations.

6.2. Graph-based diversity algorithm

To address diversity, two approaches were taken: a graph-based approach that leverages community detection and a formula-based approach that leverages the list of best recommendations. The graph-based algorithm utilizes the connectivity of the graph and the relationships between restaurants. The first step involves creating a graph in which restaurants are connected by weighted edges. The weights of the edges are the inverse of the Euclidean distance, meaning that higher Euclidean distances correspond to lower edge weights. The edges are then filtered to keep only the two heaviest edges for each restaurant, representing the two closest restaurants. An example of the resulting graph for Lausanne can be seen in Figure 6.1.

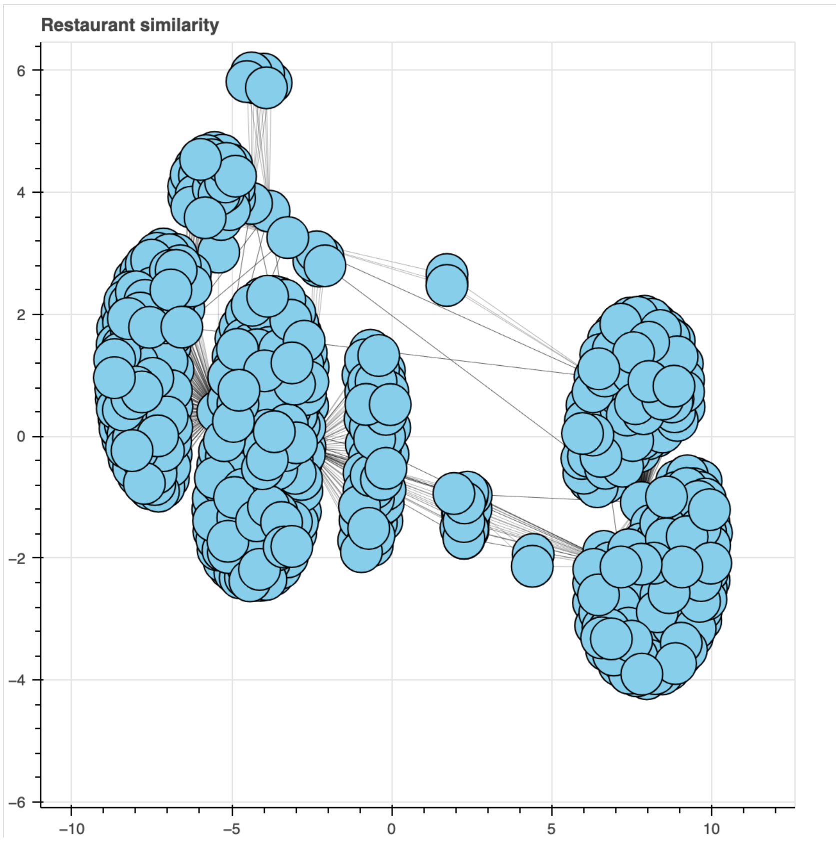


Figure 6.1: Restaurant graph for Lausanne where nodes are restaurants and edges are restaurant distances

In the graph, distinct communities can be visually identified. The next step is to detect these communities using the Louvain method. Once the communities have been identified, the PageRank of each restaurant within each community is calculated, and the highest PageRank for each community is chosen as the representative. As before, the personalized recommendation algorithm is run, and the results are sorted according to the custom metric. The first restaurant is chosen, and its community is identified. Based on the desired level of diversity, the algorithm then recommends the representative of a neighboring community. The larger the requested diversity, the farther the selected community will be.

6.3. Formula-based diversity algorithm

The formula-based algorithm works differently. First, the standard recommendations for a user are calculated. The diversity quantities for each recommendation are then determined. User preferences can be used to weight the desired diversity level for each recommendation and recalculate the top N recommendations based on both similarity and diversity scores. The current metric used to measure the accuracy of a recommendation can also be interpreted as measuring the similarity of a suggestion to the user, so the complementary score is the dissimilarity (and therefore, the diversity). Therefore, the diversity of recommendation i can be calculated as:

$$diversity_i = 1 - accuracy(recommendation_i, user)$$

To maintain a certain level of diversity within the top N recommendations, the algorithm selects a subset of N that keeps the average diversity score, D, above a specified threshold. This helps prevent overfitting in suggestions and ensures that the top N recommendations are diverse enough. The average diversity score of the top N recommendations is calculated as the average pairwise diversity:

$$D = \frac{2}{N(N-1)} * \sum_{j=1}^{N} \sum_{i>j}^{N} 1 - similarity(r_i, r_j)$$

where r_i is a recommendation and N is the total number of recommendations. This formula was proposed by Matevž Kunaver [8] in his summary of work on diversity recommendation up until 2017. This method relies heavily on the similarity metric outlined earlier, which will be thoroughly tested through online evaluation at a later stage.

6.4. Results and analysis

Despite initial promise, both methods ultimately failed to achieve the desired level of diversity. For the graph-based approach, it was not always possible to attain a low level of diversity for a user based in Lausanne, as the highest PageRank of the nearest community was already too diverse. This may be due to the fact that the concept of distance within communities in a graph is not always relevant, and community detection is generally more useful for determining membership in a community rather than estimating distances. On the other hand, the formula-based algorithm did not allow the user to reach high levels of diversity, as calculations were performed on the top recommendations for the user, which typically had an accuracy of at least 50%.

6.5. Next steps

The next steps consist in implementing a simpler graph-based method. The approach would involve starting with the same graph as the current graph-based method and allowing the desired level of diversity to determine the number of ”hops” to take in the graph starting from the top recommendation with no diversity. This approach allows for proximity to the initial recommendation with 0% diversity, while also allowing for an unlimited level of diversity by enabling the ability to walk the graph indefinitely. One idea for deciding which neighbour to hop to is to use a strategy similar to projected gradient descent, where the gradient is the variance vector of the user and the projection is onto the closest neighbour to the newly obtained vector. The step size would need to be determined.

7
Conclusion

In this work, a recommendation system for restaurants was developed using a web scrapped dataset of restaurants in Switzerland. The system was evaluated using a custom metric and several approaches were explored, including individual recommendation, group recommendation, and diversity recommendation. The results of these approaches were analyzed and key findings were discussed. In addition, next steps for improving the recommendation system were suggested.

7.1. Key findings

A custom evaluation metric was developed to adequately capture the subjective nature of food preferences and the overall dining experience. The performance of the recommendation systems was evaluated offline, using a dataset of user-restaurant interactions scrapped from the web. Several algorithms were implemented and compared, including user-based collaborative filtering, item-based collaborative filtering, and graph-based and formula-based approaches for incorporating diversity into the recommendations. The results showed that the system was able to provide accurate recommendations to individual users, with an average accuracy of 90.54%. The system was also able to provide accurate recommendations to groups of users, with an average accuracy ranging from 82% to 92% while still remaining heavily context-related. On the diversity side, the graph-based approach for diversity recommendation rose some interesting ideas, but was limited by the resolution of the detected communities. The formula-based approach was on the other hand limited by the lack of diversity in the top recommendations themselves.

7.2. Future work

Overall, the custom evaluation metric and the various recommendation approaches developed in this report demonstrate promising results for improving the accuracy and diversity of restaurant recommendations in order to ameliorate consumer experience.

There are several areas where this work could be improved and extended in the future. One possibility would be to incorporate additional data sources, such as reviews or menus, in order to provide more information to the recommendation systems. Another possibility would be to experiment with different recommendation algorithms, such as matrix factorization or neural networks, in order to see if they can achieve better performance. In addition, it would be interesting to conduct online testing of the recommendation systems in order to get a more accurate assessment of their effectiveness. Finally, it would be interesting to explore ways of incorporating diversity into the recommendation process in a more seamless and integrated manner, such as by using techniques like multi-objective optimization or by developing more sophisticated diversity metrics.

References

[1] "A Preference-Based Restaurant Recommendation System for Individuals and Groups." In: (2015). URL: https://www.cs.cornell.edu/~rahmtin/Files/YelpClassProject.pdf.

[2] Jianqiang Wang Chenbin Zhang Hongyu Zhang. "Personalized restaurant recommendation method combining group correlations and customer preferences". In: *Information sciences* 454-455.2 (2018), pp. 128–143. URL: https://www.sciencedirect.com/science/article/abs/pii/S0020025518303281.

[3] Becky Foley. *Review transparency report*. 2021. URL: https://www.tripadvisor.co.uk/TransparencyReport2021.

[4] Erik Hallström. "Group Recommender System for Restaurant Lunches". In: (2013). URL: http://www.diva-portal.org/smash/get/diva2:700100/FULLTEXT01.pdf.

[5] Salvatore Carta Ludovico Boratto. "Group recommendations with rank aggregation and collaborative filtering". In: *RecSys '10: Proceedings of the fourth ACM conference on Recommender systems* 4 (2010), pp. 119–126. URL: https://www.researchgate.net/publication/221140866_Group_recommendations_with_rank_aggregation_and_collaborative_filteringhttps://www.researchgate.net/publication/221140866_Group_recommendations_with_rank_aggregation_and_collaborative_filtering.

[6] Salvatore Carta Ludovico Boratto. "SART: group recommendation approaches for automatically detected groups". In: *International Journal of Machine Learning and Cybernetics* 6 (2015), pp. 953–980. URL: https://link.springer.com/article/10.1007/s13042-015-0371-4.

[7] Salvatore Carta Ludovico Boratto. "State-of-the-Art in Group Recommendation and New Approaches for Automatic Identification of Groups". In: *Studies in Computational Intelligence* 324 (2011), pp. 1–20. URL: https://link.springer.com/chapter/10.1007/978-3-642-16089-9_1.

[8] Tomaž Požrl Matevž Kunaver. "Diversity in recommender systems – A survey". In: *Knowledge-based systems* 123 (2017), pp. 154–162. URL: https://www.sciencedirect.com/science/article/abs/pii/S0950705117300680.

[9] Baiju Muthukadan. *Selenium with Python*. 2018. URL: https://selenium-python.readthedocs.io/#selenium-with-python.

[10] HIMANSHU PODDAR. *Zomato Bangalore Restaurants*. 2018. URL: https://www.kaggle.com/datasets/himanshupoddar/zomato-bangalore-restaurants.

[11] Leonard Richardson. *Beautiful Soup Documentation*. 2020. URL: https://www.crummy.com/software/BeautifulSoup/bs4/doc/#porting-code-to-bs4.

[12] Lucas vinh Tran. "Group Recommender Systems". In: (2021). URL: https://github.com/lucasvinhtran/group-recommender-systems.

[13] Yelp. *Yelp Open Dataset*. 2018. URL: https://www.yelp.com/dataset.

[14] Pearl Pu Yu Chen. "CoFeel: Using Emotions for Social Interaction in Group Recommender Systems". In: (2012). URL: https://ceur-ws.org/Vol-891/interfacers12_submission_4.pdf.