Recipe Recommendation Platform – Team 169 Final Report

Introduction and Motivation

This project developed a recipe recommendation platform that provides users with recommendations that have similar characteristics to recipes they enjoy yet provides opportunities to cook new recipes. These new recipe recommendations can differ in cuisine, ingredients, cooking techniques, and recipe user communities.

Problem Definition

Existing recipe recommendation platforms have yet to investigate a recommendation algorithm that leverages user history and input to provide recommendations that are categorically different from an individual's conventional choices.

Literature Survey

There are two primary approaches used today to provide individuals with recipe recommendations. Firstly, many recipe recommendation platforms focus on ingredient input to generate suggested recipes based upon ingredient/recipe similarity calculations [1, 2, 3, 4, 8, 9, 11, 13, 14, 15, 17]. These platforms that focus on understanding and returning the most similar recipes by ingredients tend to not personalize the choices for an end-user [6, 7, 8, 11]. Secondly, there are approaches that leverage user data such as favorite ingredients, dietary restrictions, and user feedback to provide a curated list of recipes and personalize the search process [5, 10, 12, 14, 15, 16, 18, 19, 20]. The common issue encountered with personalized recipe recommenders is that the user has difficulty adjusting the initial recommendations and refining the list of recommended recipes [5, 6, 7, 10, 12, 16].

Proposed Method - Intuition

The team's recommendation platform will be novel in two ways. Firstly, an individual user is given graphs and data points that show how their historical cooking tendencies (recipes, cuisines, ingredients, and techniques) are statistically related to the broader user base. For example, a frequency histogram illustrates how often an individual uses a certain cuisine relative to the general user. This approach to user analytics in the recipe recommendation space is unique because existing platforms only provide analytics about individual user habits and do not include relative population metrics. The provided visualizations will allow users to more easily understand how other people are using their kitchen and should drive motivation to explore different cuisines, ingredients, techniques, and recipe communities. Secondly, the platform's user-interface can receive continuous input from users and return recipes that align with what an individual wants to explore differently among their common cuisines, ingredients, techniques, and communities.

Proposed Method - Description

Data Input and Preparation

The team sourced its dataset from kaggle.com and the dataset included three underlying data tables. The first data table includes approximately 200,000 recipes scraped from food.com. Each record in the recipe data table provides a recipe's name, ingredients, techniques, food.com search tags, and other less relevant recipe attributes. The second data table is an ingredient map that helps classify ingredients that are similarly named to reduce the overall number of ingredients across recipes. The third data table contains approximately 700,000 recipe reviews from food.com users. Each record in the recipe review data table includes a user id, a list of recipe ids, and a list of corresponding 1-5 review scores.

The team performed many cleansing procedures upon these three data tables to ultimately create two new data tables. A few examples are as follows. Firstly, the team standardized ingredient naming across the detailed recipe data table using the provided ingredient map. Secondly, a significant percentage of recipes in the first data table had missing techniques that the team backfilled using recipe steps. Thirdly, the food.com search tags were extracted from the recipe data table to create individual attributes like 'cuisine' and 'meal of the day'. Fourthly, the team only kept recipes from the first data table that had at least 2 reviews and an average rating of 4 or higher to filter out any negative user interactions. Lastly, the recipe reviews data set was transformed to have each record be a separate combination of the user_id, recipe_id, and 1-5 rating.

The resulting two data tables are known as 'recipes_in' and 'users_in'. The 'recipes_in' data table contains a cleansed version of the previously mentioned recipe attributes table, using a 'recipe_id' as the table's primary key. The 'users_in' data table is the resulting transformation of the raw recipe reviews data set, having the 'user_id' and 'recipe_id' as primary keys. These two data sets are the foundation of the recommendation platform and enable the team to construct all the underlying data tables used in intermediary calculations throughout the team's algorithm.

User Descriptive Analysis

The User Descriptive Analysis (UDA) portion of the recipe recommendation platform determines the frequency of specific cuisines, ingredients, and techniques used across a user's recipe history and how it compares to the overall population. The UDA process leverages the previously mentioned 'recipes_in' and 'user_in' data tables to calculate these frequency counts for a given user and the overall population across the three categories. The team visualizes these comparisons in the platform through three histograms. Each histogram depicts the user frequency counts for a specific category in one color and uses another color to show the distribution of frequency counts across the entire user base. These visualizations can provide users insight on what to explore in new recipes across these three categories.

Similarity Measurement

Similarity measurements in the recommendation platform are used to develop an algorithm to measure the cuisine, ingredient, and technique similarity scores between each recipe in the data set and a given user's preference (based on the user's top-rated recipes). New recipes can then be recommended based on how a user wants to vary cuisine, ingredients, and techniques. Ingredient similarity is calculated using a combination of cosine similarity and a natural language processing technique called Word2Vec. Word2Vec is used to transform an individual recipe's ingredients into a vector. First, the team will sum all the ingredient vectors in the user's top-rated recipes to create a 'user ingredient preference' vector. From there, cosine similarity is calculated between each recipes' ingredient vector and the 'user ingredient preference' vector [11,13]. The team will subsequently calculate a 'user cuisine preference' and a 'user technique preference' vector, applying the same methodology across every available recipe, to find the respective cuisine and technique similarity scores.

Community Analysis

The team performed an analysis on a network graph of recipes in our data set, where linkages between data sets are defined by the percentage of the two recipes' total user base that is shared by the recipes. The team leverages this network graph in two ways. First, communities of recipes detected from the graph show users where their top recipes fit into the larger user base. Secondly, a distance measure between the network of recipes is used as a similarity measurement within the algorithm used to rank recipe recommendations for users. For community detection, the team utilizes the Louvain algorithm due to its computational efficiency and quality of identified communities. For distance measurement, the team has 'upscaled' the graph to the community level to compute the distances between communities instead of recipe to recipe given gaps where recipes are not connected.

Recommendations and Visualizations

The recipe recommendation platform was developed using Dash, a python framework created by plotly that is used for creating interactive web applications. Dash was chosen for the recommendation platform because of its performance, documentation, and aesthetics. The team tested Heroku, a cloud hosting service, to host the web application. Initially, Heroku performed well and provided a simple and effective way to access the web application. Unfortunately, as more computational requirements were introduced the platform exceeded the limits of the free hosting service Heroku provided. The team chose to host the platform locally but a paid subscription to Heroku to host a future iteration of the platform would be easy to obtain and roll out.

The recommendation platform is comprised of two different interactive pages titled 'User Summary' and 'Try New Recipes'. The 'User Summary' page is comprised of visualizations on a given user's historical recipe use and how this user's history relates to the broader user base. The previously mentioned User Descriptive Analysis portion of the recommendation platform will be the foundation of the 'User Summary' page. The 'Try New Recipes' page provides users with the functionality to generate new recipe recommendations based on several pre-selected filters and the degree in which a user wants to explore different cuisines, ingredients, techniques, and communities.

The team implemented sliding bars within the 'Try New Recipes' page to indicate a user's preference for similarity (a value of +1), dissimilarity (a value of -1), or any value in between for each category in a new recipe recommendation set. After applying the user's pre-selected filters, an aggregate similarity score is calculated for each potential recipe that is weighted based on the user's input for cuisine, ingredients, techniques, or recipe community. Recipes with the top aggregate scores across the four category inputs will be displayed to the user. To allow for aggregate scoring across all four similarity measurements, the similarity scores for each category input across all recipes are linearly scaled between –1 and 1. If dissimilarity is preferred in a category, a negative similarity measurement will contribute positively to the aggregate score. If similarity is preferred in a category, a positive similarity measurement will contribute positively to the aggregate score.

Experiments and Evaluation

The following section contains the team's testbed and the test results used to address each question.

1. **How well does the platform performance scale with users of varying recipe history size?**
    a. **Test:** Evaluate the computational speed for recipe recommendations by filtering on users whose individual data records are of varying size.

i. **Results:**

1. Both pages in the recommendation platform load in less than two seconds and there is negligible difference in speed between the largest user and average times of 30 random users. These tests were run locally on a MacBook Pro. [22]

2. **Were the algorithm approaches within the Similarity Measurement and Community Analysis components appropriate and effective?**

a. **Test:** Evaluate the trained Word2Vec model's ability to provide logical ingredient similarity scores.

i. **Results:**

1. The team evaluated the validity of the Word2Vec model's ingredient similarity output by utilizing the .most_similar() function within the gensim.models package. The function is used by inputting test ingredients and reviewing the output of most similar ingredients based on cosine similarity calculations. After testing multiple ingredient inputs, the team concluded the Word2Vec trained model was accurately returning similarity scores based on a given input. [23]

b. **Test:** Evaluate community detection algorithm (e.g. Louvain) based on output network modularity, performance, and cluster size/count.

i. **Results:**

1. The team evaluated a combination of filtering and sampling methods with both the Louvain and Greedy Community Detection algorithms. The Greedy Community Detection Algorithm could not be run locally even with a filtered data set; thus, the Louvain algorithm was chosen. The team determined that the Louvain algorithm was best ran on this data set when edges must have at least 8% of the user base share a recipe. [21]

c. **Test:** Determine approach to backfill community assignments based on output cluster sizes.

i. **Results:**

1. The initial community assignments provided ~95% of required cluster assignments. The remaining assignments were made using a k-nearest neighbors-like algorithm which took the majority classification from the nearest 10 neighbors, where the distance was the percentage of shared users between the two recipes.

d. **Test:** Evaluate individual similarity scores based on user ratings of individuals who rated at least 500 recipes (~200 users). Use pseudo-CV technique to hold out user recipes, then understand if highly rated recipes for a user received high similarity scores and vice versa.

i. **Results - Ingredient Similarity Testing:**

1. For each user, we used their 'user ingredient preference' vector and calculated ingredient similarity scores across all recipes, ranking them largest to smallest. The team averaged ingredient to ingredient similarity scores between all ingredients in the Top 20 and Bottom 20 similarly

ranked recipes and compared them to the ingredients in the user's top-rated recipes. The team determined the ingredient similarity calculations were working as expected due to the average ingredient to ingredient similarity score in the Top 20 recipes being higher than that of the Bottom 20 recipes.

ii. **Results - Cuisine Similarity Testing:**

1. The team applied the previously mentioned ranking methodology to assign cuisine similarity scores for all recipes in the dataset. The team determined that the most frequent cuisine in a user's Top 20 similar recipes equaled the most frequent cuisine in a user's top-rated recipes.

iii. **Results - Technique Similarity Testing:**

1. The team applied the previously mentioned methodology to assign technique similarity scores for all recipes in the dataset. The team determined that the ten most frequently appearing techniques in the Top 20 similarly ranked recipes were in line with the ten most frequently appearing techniques in a user's reviewed recipes. In addition, the ten most frequently appearing techniques in the Bottom 20 similarly ranked recipes were not in line with the top techniques in the user's reviewed recipes.

iv. **Results - Community Similarity Testing:**

1. The team created a visualization that plots the percentage of users that share a particular recipe and the community similarity score between those two recipes [24]. The visualization supports the claim that the community similarity scores will be higher for recipes that are closer within the community network graph.

3. **Does each functional component of the final product work as intended?**
   a. **Test:** Developed testing script for each team member to walk through step by step to validate the platform functions as expected.
      i. **Results:** The team identified opportunities to improve the final product, and each of the following fixes have been implemented within the platform:
         1. The 'user ID' field initially did not persist when changing tabs, making things confusing for a user who wants to go from 'User Summary' to 'Try New Recipes' without re-entering their ID.
         2. The displayed names for techniques, cuisine, and recipe titles were not appropriately capitalized or punctuated, leading to an unprofessional look.
         3. The community detection scores were inverted, so the functions of 'Similar' and 'Dissimilar' were flipped.
         4. Axis labels were missing
         5. Colors in the data visualizations were too light
4. **How interactive is the user-interface and does the platform return recommendations as expected?**

a. **Test:** The team conducted mock-user tests to confirm that the results in the final product were as expected. Each team member was assigned 40 randomly selected user IDs, and every unique combination of dimension similarity (ingredients, techniques, cuisine, and community) was evaluated twice. For the test to succeed, the resulting visualizations needed to show that the recommendations were matching the inputs of 'Similar', 'Dissimilar', or 'Doesn't Care'.
    i. **Results:** The team identified opportunities to improve the final product, and each of the following fixes were implemented within the platform:
        1. The callback function which updated similarities was running in parallel to updates to the user ID, causing recommendations to lag changes to the user ID field.
        2. Inter-community distances were not entering the algorithm, so selecting 'Similar' for 'Community' was not producing recommendations for recipes that lay in the user's most populated communities.
        3. Recipes' punctuation was missing for possessive nouns (e.g. "Mom's Lasagna").

Conclusions and Discussion

The team's goal for this project was to provide a user-friendly platform that allows people to explore a world of new recipes in a way that they hadn't experienced yet. Conventionally, recipe recommendation platforms are aimed to provide recommendations that a user would like based on a multitude of different factors. This project does not guarantee that an individual user will enjoy their recipe recommendations yet achieves the goal of providing a platform to explore new dimensions of the culinary world. The team did not conduct user-testing on the recommendation outputs because it was outside the scope of the project's goal. Instead, the team validated each individual component of the algorithm to ensure that if a user wanted to explore different recipes along our four dimensions, the results were appropriate.

Each team member contributed equal amounts of effort throughout the duration of the project.

Philip Murray (POC): Technical writing lead for proposal, progress, and final report deliverables. Led wireframe development for platform user-interface.

Ying Sui: Led exploratory data analysis and her insights contributed to algorithm development. Led initial creation of user-interface and continued efforts here through project completion.

Matthew Maliniak: Led data extraction and cleansing for algorithm and visualization development. Collaborated with Ying throughout the remainder of the final user-interface build.

Andrew Taylor: Project Manager who led weekly meetings and managed workload. Led the development of Community Analysis, cuisine similarity, and aggregate scoring methodologies. Assisted with experimentation and evaluation of the final recommendation platform.

Qianwen Zhang: Led the development of ingredient and technique similarity scoring methodologies. Led experimentation and evaluation efforts for the recommendation platform.

References

1. A. Nadamoto, S. Hanai and H. Nanba, "Clustering for Similar Recipes in User-Generated Recipe Sites Based on Main Ingredients and Main Seasoning," 2016 19th International Conference on Network-Based Information Systems (NBiS), 2016, pp. 336-341, doi: 10.1109/NBiS.2016.49.
2. A. Morales-Garzón, J. Gómez-Romero and M. J. Martin-Bautista, "A Word Embedding-Based Method for Unsupervised Adaptation of Cooking Recipes," in IEEE Access, vol. 9, pp. 27389-27404, 2021, doi: 10.1109/ACCESS.2021.3058559.
3. Cambridge, Peter Forbes University of, et al. "Content-Boosted Matrix Factorization for Recommender Systems: Proceedings of the Fifth ACM Conference on Recommender Systems." ACM Conferences, Association for Computing Machinery, 1 Oct. 2011, https://dlnext.acm.org/doi/abs/10.1145/2043932.2043979.
4. Min, Weiqing, et al. "A Delicious Recipe Analysis Framework for Exploring Multi-Modal Recipes with Various Attributes: Proceedings of the 25th ACM International Conference on Multimedia." ACM Conferences, Association for Computing Machinery, 1 Oct. 2017, https://dl.acm.org/doi/10.1145/3123266.3123272.
5. Ueda, Mayumi, et al. "User's Food Preference Extraction for Personalized Cooking Recipe Recommendation: Proceedings of the Second International Conference on Semantic Personalized Information Management: Retrieval and Recommendation - Volume 781." Guide Proceedings, CEUR-WS.org, 1 Oct. 2011, https://dl.acm.org/doi/10.5555/2887675.2887686.
6. Lin CJ., Kuo TT., Lin SD. (2014) A Content-Based Matrix Factorization Model for Recipe Recommendation. In: Tseng V.S., Ho T.B., Zhou ZH., Chen A.L.P., Kao HY. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2014. Lecture Notes in Computer Science, vol 8444. Springer, Cham. https://doi.org/10.1007/978-3-319-06605-9_46
7. Elsweiler, David, et al. "Exploiting Food Choice Biases for Healthier Recipe Recommendation." Northumbria Research Link, The 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 7 Aug. 2017, https://nrl.northumbria.ac.uk/id/eprint/30863/.
8. Li, Shuyang, et al. "SHARE: A System for Hierarchical Assistive Recipe Editing." ArXiv.org, 17 May 2021, https://arxiv.org/abs/2105.08185.
9. Su, Han, et al. "Automatic Recipe Cuisine Classification by Ingredients: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication." ACM Conferences, Association for Computing Machinery, 1 Sept. 2014, https://dl.acm.org/doi/abs/10.1145/2638728.2641335.

10. Elsweiler, David, et al. “Exploiting Food Choice Biases for Healthier Recipe Recommendation: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval.” ACM Conferences, Association for Computing Machinery, 1 Aug. 2017, https://dl.acm.org/doi/10.1145/3077136.3080826.
11. Teng, Chun-Yuen, et al. “Recipe Recommendation Using Ingredient Networks: Proceedings of the 4th Annual ACM Web Science Conference.” ACM Conferences, Association for Computing Machinery, 1 June 2012, https://dl.acm.org/doi/abs/10.1145/2380718.2380757.
12. H. Xie, L. Yu and Q. Li, "A Hybrid Semantic Item Model for Recipe Search by Example," 2010 IEEE International Symposium on Multimedia, 2010, pp. 254-259, doi: https://ieeexplore.ieee.org/abstract/document/5693849.
13. Dangeti, Pratap. “Statistics for Machine Learning.” O'Reilly Online Learning, Packt Publishing, July 2017, https://learning.oreilly.com/library/view/statistics-for-machine/9781788295758/eb9cd609-e44a-40a2-9c3a-f16fc4f5289a.xhtml.
14. K. R. Pawar, T. Ghorpade and R. Shedge, "Constraint based recipe recommendation using forward checking algorithm," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016, pp. 1474-1478, doi: 10.1109/ICACCI.2016.7732256.
15. T. L. Cheng, U. K. Yusof and M. N. A. Khalid, "Content-based filtering algorithm for mobile recipe application," 2014 8th. Malaysian Software Engineering Conference (MySEC), 2014, pp. 183-188, doi: 10.1109/MySec.2014.6986011.
16. Z. Li, J. Hu, J. Shen and Y. Xu, "A Scalable Recipe Recommendation System for Mobile Application," 2016 3rd International Conference on Information Science and Control Engineering (ICISCE), 2016, pp. 91-94, doi: 10.1109/ICISCE.2016.30.
17. Yamanishi, Ryosuke, et al. “Alternative-Ingredient Recommendation Based on Co-Occurrence Relation on Recipe Database.” Procedia Computer Science, Elsevier, 1 Sept. 2015, https://www.sciencedirect.com/science/article/pii/S1877050915022656.
18. Sema Akkoyunlu, Cristina Manfredotti, Antoine Cornuéjols, Nicolas Darcel, Fabien Delaere. Exploring eating behaviours modelling for user clustering. HealthRecSys@RecSys 2018 colocated with ACM Recsys’18 (ACM Conference Series on Recommender Systems), Oct 2018, Vancouver, Canada. pp.46- 51. ffhal-02480562f
19. Yalvaç, Fulya, et al. “Social Recipe Recommendation to Reduce Food Waste: Chi '14 Extended Abstracts on Human Factors in Computing Systems.” ACM Conferences, Association for Computing Machinery, 1 Apr. 2014, https://dl.acm.org/doi/abs/10.1145/2559206.2581226?casa_token=M2lklDvkAkAAAAAA%3Ad

5tOs9EJyAgVfG-AxPlgXFyhpvv0DekqhZ1lDGJTsjdyOHrBS-CbMdyAHHkjGqKkjeT93737_wXrJH8.

20. Freyne, Jill, and Shlomo Berkovsky. "Intelligent Food Planning: Proceedings of the 15th International Conference on Intelligent User Interfaces." ACM Conferences, Association for Computing Machinery, 1 Feb. 2010, https://dl.acm.org/doi/abs/10.1145/1719970.1720021?casa_token=hRF8wDwAVKkAAAAA%3ACDw8VTP7d3dhRAquWbJT7uTBJXzSRmkHLbsvkawx-Clldn7jCig3rrqJkLSQ5gthzpMcugtpDW1ebuM.

21. Community Detection Algorithm Testing Results

Filter/Sample Method	Community Detection Algorithm	Number of Communities Detected	Modularity	Coverage	Performance	Community Subgraphs	Community Evenness	Runtime
Full graph	Louvain							Inf
>2 Shared Users	Greedy Community Detection							Inf
>2 Shared Users	Louvain	There are 317 communities	0.392159984	0.555	0.692	81	Highly uneven	~20 min local
>8% shared users	Louvain	There are 32 communities, but only 17 excluding disconnected subgraphs with 35 recipes (<.01%)	0.477323697	0.531	0.785	15	Three large, others even (apart from subgraphs)	~20 min on AWS
50% uniform node sample	Louvain	There are 18 communities	0.473945223	0.519	0.81	2	Three large, others even (apart from subgraphs)	~20 min local
75% uniform node sample	Louvain	There are 16 communities	0.476063036	0.527	0.825	1	Three large, others even (apart from subgraphs)	~20 min local
90% uniform node sample	Louvain	There are 17 communities	0.47454791	0.528	0.798	1	Two large, others even	~20 min on AWS

22. Performance Testing w/ Different Users

USER SUMMARY	Average Time (s)	Time for Largest User
Ingredients chart	0.26	0.27
Ingredients wordcloud	1.15	1.29
Techniques chart	0.14	0.15
Cuisine chart	0.11	0.07
Community chart	0.68	0.58
TRY NEW RECIPES		
Recommendations	0.08	0.04
Wordclouds	0.54	0.53
Techniques chart	0.02	0.01
Cuisine chart	0.01	0.01
Community chart	0.06	0.07

23. Word2Vec Testing, Ingredient Input: ‘tomato paste’

```
[('tomato puree', 0.8048397302627563),
 ('tomato sauce', 0.7564524412155151),
 ('crushed tomatoes', 0.7190336585044861),
 ('whole tomatoes', 0.7148817777633667),
 ('bay leaves', 0.7098484039306641),
 ('canned tomatoes', 0.7092584371566772),
 ('tomato juice', 0.7071650624275208),
 ('dry red wine', 0.6920954585075378),
 ('bay leaf', 0.6816862225532532),
 ('whole canned tomatoes', 0.6762807965278625)]
```

24. Community Similarity Score Validation