Final Report: Recipe Recommender

1. Introduction

Background

In the vast landscape of culinary choices, the search for the perfect recipe that aligns with individual tastes and preferences can be both exciting and overwhelming. This is where the importance of a sophisticated recipe recommender system becomes evident. Imagine a tool designed to curate a culinary journey just for you, considering your unique palate, dietary preferences, and cooking style.

In a world where time is precious and the culinary possibilities are endless, a recipe recommender system becomes a culinary companion, streamlining the process of finding recipes tailored to your liking. By leveraging a rich dataset comprised of a diverse array of recipes and user reviews, this system empowers you to discover new flavors, experiment with confidence, and transform ordinary meals into extraordinary experiences.

Problem Statement

Develop a robust recipe recommender system leveraging a comprehensive dataset of recipes and a corresponding dataset of user reviews. The goal is to enhance the culinary experience for users by creating a personalized recommendation engine that suggests recipes based on individual preferences and past review history.

2. Datasets

Recipes dataset

1. Name: Recipe name
2. Id: recipe id
3. Minutes: recipe prep time in minutes
4. Contrubuter_id: user id
5. Submitted: Date of recipe submission
6. Tags: List of keywords associated with the recipe
7. Nutrition: List of values corresponding to different nutritional elements
8. N_steps: number of steps
9. Steps: List of recipe steps
10. Description: recipe description

Users dataset

1. user_id: Unique user id
2. recipe_id: unique recipe id
3. date: Date of review submission
4. rating: Numeric rating between 0-5

5. review: written review of the users

3. Data cleaning and wrangling

The recipes dataset encompasses 231,637 records distributed across 12 fields. Notably, there were no missing values identified in pertinent columns, although the description column did contain null values; however, it was not utilized in the model. To enhance the model's capabilities, I introduced several novel features. The nutrition column was divided into seven distinct fields, each dedicated to a specific nutritional aspect. Employing Natural Language Processing (NLP), I derived 'is_vegan,' 'is_vegetarian,' and 'cuisine' fields. The former two are Boolean in nature, indicating whether a recipe is vegan or vegetarian, while the 'cuisine' field encompasses values such as North American, European, African, Asian, Australian, and South American. Additionally, I transformed numerical data from the minutes, n_steps, and n_ingredients columns into categorical columns, classifying values as short, medium, or long. This comprehensive approach to feature engineering contributes to the richness and precision of the model's predictive capabilities.

The users dataset comprises 11,322,367 records distributed across four fields. Notably, the dataset required no data wrangling, except for the review column of string type, which contained null values; however, the column was excluded from the modeling process.

4. Exploratory Data Analysis

Preliminary look

The users dataset exhibits an imbalance in the rating column, where the majority of users predominantly assign ratings of 4 or 5, as illustrated in Figure 1. Additionally, Table 2 reveals a noteworthy percentage of users who have provided only one recipe review in the dataset. This observation underscores the necessity to address the cold start problem within the recommendation engine. Effectively addressing this issue becomes pivotal for ensuring robust and accurate recommendations, particularly for users with limited engagement history.

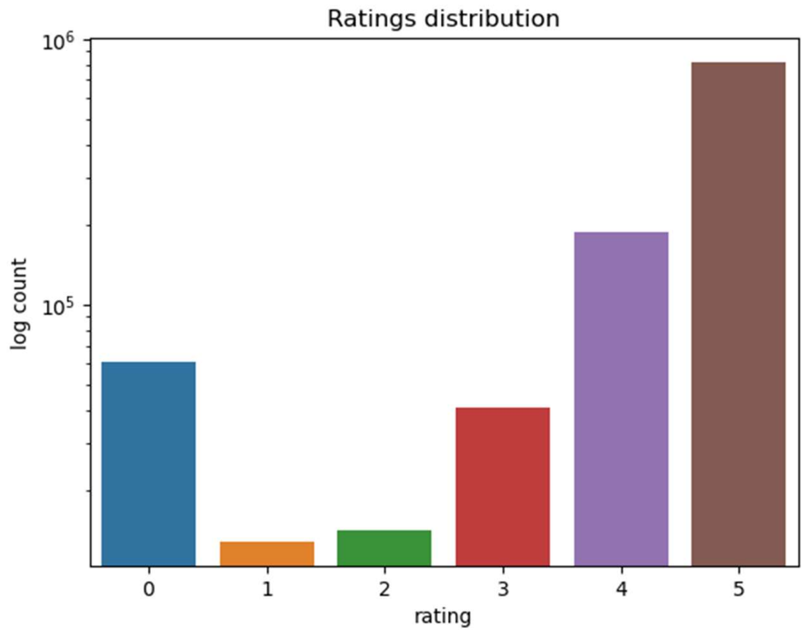

Figure 1. User ratings distribution

Ratings per user	Number of users
0-20	412819
21-40	87359
41-60	50642
61-80	41145
81-100	29916
101-200	94028
201-300	57299
301-400	48813
401-500	38360
501-1000	97148
1001-2000	95852
2000+	78986

Table 1. Ratings per user distribution

Ratings per user	Number of users
1	166256
2	45476
3	28038
4	20576
5	17105
6	15396
7	13559
8	11688
9	10935
10	10400
11	9262
12	9060
13	8268
14	7728
15	7545
16	6960
17	6222
18	6210
19	6175
20	5960

Table 2. Ratings per user distribution (1-20)

The cuisine feature, derived through NLP analysis of the tag field for each recipe, unveils North American, European, and Asian cuisines as the most prominent types, as depicted in Figure 2. This insight highlights the prevalence of these culinary styles within the dataset, providing valuable information for understanding the diverse preferences and trends reflected in the recipes.

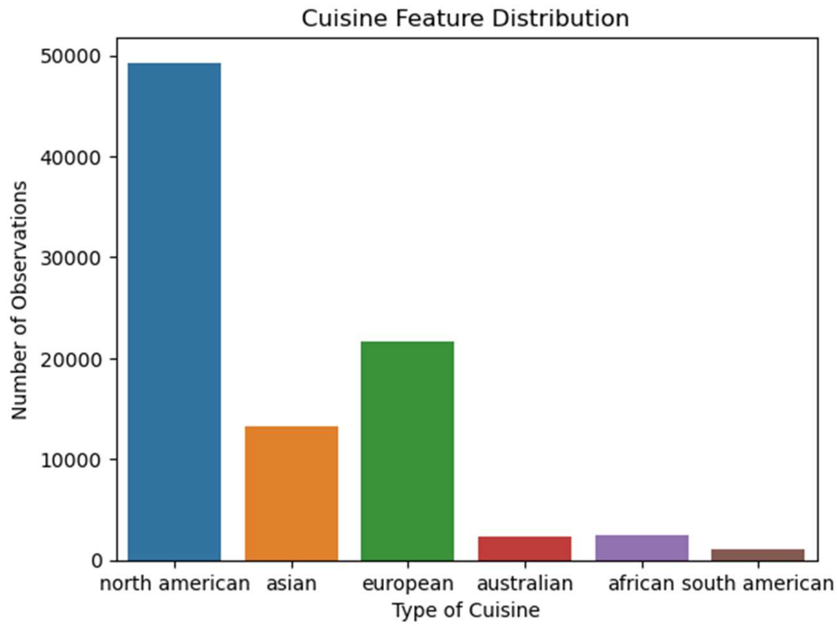

Figure 2. Data distribution for the cuisine feature visualized through a histogram.

5. Modeling

I implemented three key models to enhance the recommendation system:

1. Simple Recommender: This model identifies the top-n highly rated recipes by aggregating preferences across all users. It offers a straightforward approach by recommending popular recipes based on overall user ratings.
2. Content Recommender: Focused on individual recipes, this model suggests the top-n recipes that are closely related to a specific given recipe. It leverages content-based filtering, considering factors such as ingredients, cuisine, and other relevant features to find recipes with similar characteristics.
3. Hybrid Recommender: The hybrid model takes a collaborative filtering approach by recommending the top-n recipes that are reviewed by users with similar preferences to a given user. This combines both user-based collaborative filtering and content-based methods, offering personalized recommendations that align with a user's taste while considering the preferences of like-minded users.

These three models provide a well-rounded recommendation system, catering to different user needs and preferences. The simplicity of the simple recommender, the content relevance of the content recommender, and the personalization of the hybrid recommender collectively contribute to a comprehensive and effective recommendation experience.

Simple Recommender

The simple recommender identifies the top-n recipes with the highest overall user ratings, employing the IMDB weighted rating formula to calculate a composite score for each recipe. However, a potential limitation of this system is its simplicity; it overlooks individual user preferences and recipe attributes, relying solely on the cumulative ratings given by all users.

	Recipe Name	Weighted Rating	Number of Ratings
134684	mexican stack up rsc	4.965116	217
35255	caprese salad tomatoes italian marinated toma...	4.903545	52
129662	mango salsa 1	4.892966	74
207459	syrup for blueberry pancakes	4.880549	57
214172	toffee dip with apples	4.876816	55

Figure 3. Simple recommender's top 5 recommendations

Content Recommender

The content recommender functions by training the recipes dataset using a k-means model. After conducting an elbow curve analysis, I identified the optimal number of clusters to be within the range of 3 to 5, ultimately deciding on 4 clusters. Subsequently, recipe records are allocated to one of these 4 clusters, and the recommendations consist of the top-n recipes with the closest data points within the assigned cluster. While k-means may not offer the precision of individual similarity metrics like cosine similarity, it boasts lower memory requirements, making it a favorable choice for this implementation.

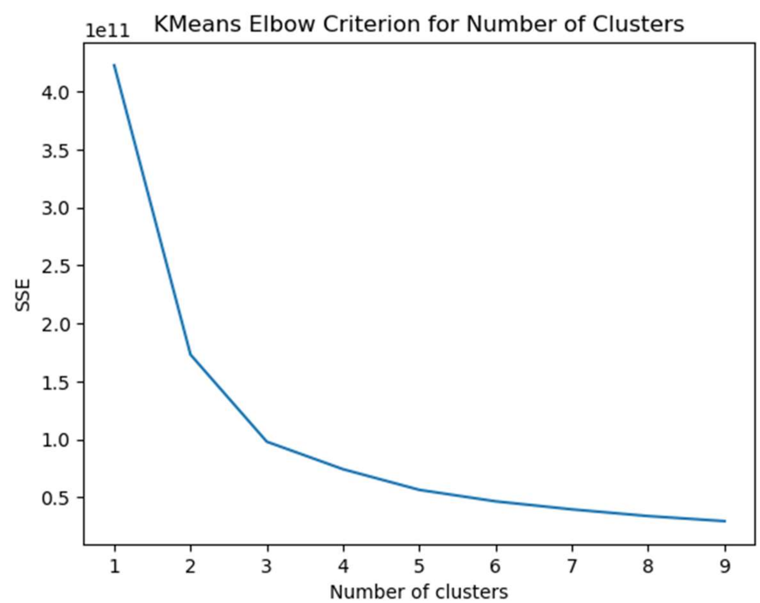

Figure 4. K-means elbow criterion curve to determine number of clusters

```
id
503239                       baked beans with baked bacon
35418      beef stuffed bell peppers with creole sauce
349048                                 enchiladas verde
33431                                          meatloaf
426424                         spicy lamb stuffed peppers
```

Figure 5. Content recommender's top 5 recommendations for a given recipe.

Hybrid Recommender

The Hybrid Recommender operates by identifying the most similar users through correlation and then calculates a weighted rating for a list of recipes reviewed by these comparable users. In the absence of similar users meeting the correlation criteria, the system resorts to providing a content recommendation based on the input user's highest-rated recipe.

recipe_id	weighted_rating	name
2625	5.0	spinach cashew salad
175343	5.0	banana spice bars
168748	5.0	rachael ray s mamacello pasta
167894	5.0	boneless pork chops milanese
167792	5.0	honey glazed corned beef

Figure 6. Hybrid Recommender's top 5 recommendations for a given user.

Model Performance

Given the absence of a dedicated performance metric for this case, the assessment of the model's performance requires a manual approach. To achieve this, I randomly select 20 users from the dataset and extract their most recently reviewed recipes. Subsequently, I conducted a manual comparison between these recipes and the hybrid recommendations generated from the entire review history of each user. The criteria for comparison include cuisine type and food category (e.g., appetizer, entree, dessert, etc.). The calculated mean relevancy score was 0.42, indicating that, on average, 2 out of the 5 recipe recommendations were deemed relevant for each user.

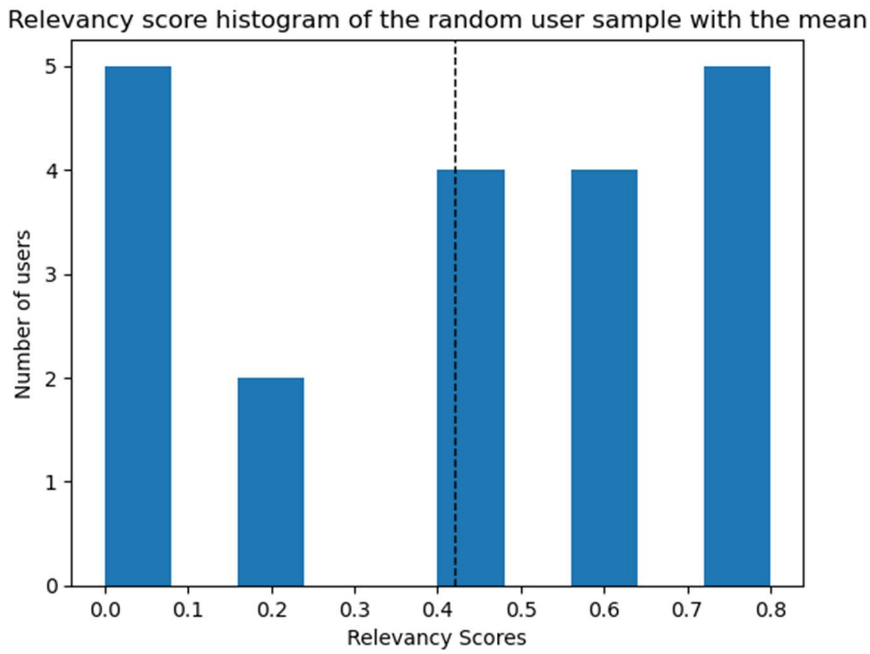

Figure 7. Distribution of relevancy score with the mean on the dotted line

6. Conclusions

The primary recommendation engine encompasses the following types: Simple, which suggests top-n rated recipes by all users; Content, which identifies top-n closest recipes to a given recipe; and Hybrid, which proposes top-n closest recipes reviewed by users most similar to a given user's recipe history. In the absence of established performance metrics, model effectiveness was evaluated manually, revealing that, on average, 2 out of 5 recommendations were considered relevant in a random sample of 20 users.

To enhance recommender performance, additional features could be extracted from the data, such as recipe type (appetizer, entrée, etc.), and a more in-depth analysis of ingredients could be conducted. Ultimately, the data significantly influences model performance, making improved datasets, especially those providing more user metadata, a key avenue for enhancing overall system performance.