RECIPE RECOMMENDATION SYSTEM

[redacted]

ABSTRACT

In this project we explore how a recommendation system can be used to recommend recipes to a user with specific requirements. The requirements of the person could be restricted in the form of the ingredients they have, time taken to make the recipe, nutritional value etc. Using the "Recipe and Interactions" dataset, from Kaggle - Food.com, we build three AI models for the recommending recipes – content-based filtering, knowledge-based filtering and collaborative filtering recommendation system, with neighbourhood algorithm (KNNMeans, KNNBase, KNNZScore) and matrix factorization (SVD, SVDpp). Of all the collaborative filtering models, SVDpp performs best when we use RMSE and MAE to test the accuracy of the recommendations. Since all these models have their own limitations, we have finally built a switching hybrid model which combines the three recommendation models.
Keywords: Recommendation system, Content-Based Filtering, Knowledge-Based Filtering, Collaborative Filtering, K-Nearest Neighbour, Mean Square Error.

1. INTRODUCTION

We can see recommendation systems extensively applied in diverse domains such as to recommend movies, books, music, etc according to users' preferences. In this study, we aim to develop a recipe recommendation system designed to assist individuals in making informed decisions regarding the utilisation of leftover ingredients, with the overarching goal of minimising food wastage.

The primary objective of this research study is to construct an AI based algorithm that facilitates users in optimizing the utilization of their available ingredients by providing customized recommendations. With our knowledge-based and hybrid models, users could use the most (or all) of their ingredients and get tailored recommendations based on the time they have, the number of steps in cooking, time taken, etc. Furthermore, our model aspires to enhance the precision of future recipe recommendations by incorporating user preferences and ratings, by identifying users who have similar preferences. To create this recommendation system with more factors, a sophisticated modelling approach had to be used. The dynamic nature of user feedback and evolving taste complicates the task of refining the model.

2. PROBLEM FORMULATION

How can we implement an AI agent that suggests recipes to individuals considering resource availability and user preferences?

3. REVIEW OF EXISTING RESEARCH

We now provide reviews of a peer reviewed article on recommendation systems along with an examination specific to our recipe recommendation data. We further reinforce the context of our study and emphasize the contributions made by prior research.

3.1. RECOMMENDATION SYSTEMS

Recommendation Systems leverage advanced algorithms to customize suggestions based on user preferences. Common challenges in modern recommendation systems include scalability, cold-start, and sparsity (Roy and Dutta [7]). Frequent approaches for recommendation systems include ML algorithms of content-based filtering, collaborative filtering, and deep learning (Roy and Dutta [1]).

Content-based recommendation systems organize data into profile categories based on different features. According to how a user rates an item, items within the same item profile are aggregated to build a user profile. Strengths of this approach include dynamic adaptation to changing user preferences, assurance of profile privacy, and techniques to overcome cold-start problems (Roy and Dutta [3]). Collaborative filtering recommendation systems find a group of users with similar likes and dislikes, known as the neighbourhood (Roy and Dutta [4]). Two types of collaborative filtering are memory-based and model-based approaches. Memory-based approaches utilize the utility matrix for prediction whereas model-based approaches utilize data mining and machine learning algorithms to create a predictive model for user ratings before predictions.

Hybrid recommendation systems combine myriad techniques into a singular approach. Meta-level, feature-augmentation, feature-combination, mixed hybridization, cascade hybridization, switching hybridization, and weighted hybridization are among the hybridization approaches highlighted in the study (Roy and Dutta [6]). This combination improves the various challenges of each individual recommendation system by maximizing the strengths in each approach.

3.2. OUR CONTRIBUTIONS

Today, recipe recommendation systems are in an increasingly high demand, with AI reshaping how people decide what to cook. Our exploration into recipe recommendations focuses on AI techniques that provide personalized suggestions tailored to individual preferences, similar users' preferences, available ingredients, and time constraints. Unlike conventional movie recommendations, the uniqueness of each recipe poses various challenges, requiring diverse recommendation approaches such as knowledge-based, content-based, and collaborative filtering methods. Our 'switching hybrid model', integrates these strategies and reflects the varying tendencies of each user. Despite challenges such as data sparsity and model interpretability, our project considered the future of recipe recommendations, integrating real world data, user feedback, and ML techniques for a modern culinary experience.

4. BACKGROUND

4.1. THE AI AGENT AND ITS TASK ENVIRONMENT

To build a model for recipe recommendation, it is vital to first understand the players we are working with in this model. More specifically, our recipe recommendation AI **agent** would face an **environment** of all possible recipes, in which the agent **perceives** what ingredients the user has (with the time they have, the efforts taken, etc) and what the user has rated in the past. The AI agent then **learns** by studying patterns in the user behaviour and then refines its **actions**, i.e., recipe recommendations.

Let's define our model under the PEAS framework. This can change and improve based on the amount of data we have. For example, if we had data about the user's food preference at different times of day, we could include that in the environment and the agent can learn from that to recommend the most suitable food item during that time of the day.

Table 1: PEAS description of the environment of the AI Agent

Agent type	Performance measure	Environment	Actuators	Sensors
Recipe Recommender	User Rating	Recipe Dataset	Suggesting recipe	Resource Constraints & user history

4.2. DATA

Utilizing the Food.com "Recipe and Interaction" dataset from Kaggle, we incorporated two key data sets into our model. The first, RAW_Recipe.csv, contains recipe details, including characteristics such as the number of steps, ingredients, nutritional value, time requirements, etc. The second, RAW_interaction.csv, comprises of user ratings for all recipes, offering insights into user preferences and interactions within the dataset. This dataset has 180K+ recipes and 700K+ recipe reviews covering 18 years of user interactions and uploads on Food.com.

5. PROPOSED SOLUTIONS

In any recommendation system, users share basic constraints, such as available ingredients, preparation time, and recipe steps. The model then uses this information to enhance recommendations. We've developed various models to cater to recipe suggestions based on user constraints:

- Knowledge-based Filtering
- Content-based Filtering
- Collaborative Filtering
    - KNN inspired algorithms (item based and user based)
    - Matric factorization based algorithms
- Hybrid Model

5.1. KNOWLEDGE BASED METHOD

The knowledge-based recommendation system gathers user inputs regarding resource availability such as ingredients to be included, time constraints, nutritional preferences etc. To further enhance the search, the model also asks the user whether they would like to receive recipe recommendations including all preferred ingredients. Using this information, the model filters the relevant recipes, satisfying the constraints. Subsequently, the model prompts the user to specify the preferred ranking criterion for the recipe recommendations, offering options such as average user ratings, calorie content, protein levels, etc. The recommendation model then arranges the resulting list by employing a weighted average of the chosen criterion. For instance, if the user opts for recipe recommendations based on protein content, the model organises the list in descending order of protein levels. In scenarios where multiple recipes exhibit similar protein content, further sorting is conducted based on the weighted average of user ratings, with weights determined by the number of ratings received per recipe. This methodology proves advantageous in instances where user history is unavailable, and individuals have preferences.

5.2. CONTENT BASED METHOD

A content-based recommendation system model operates by utilising information associated with a specific user (U1) and recipe (R1) to propose a subsequent recipe (R2). The recommended recipe (R2) is selected based on its resemblance to the recipes typically prepared and favoured by user U1. The strategy used is developing the model on the basis of the characteristics that describe the user item interactions. In this case we use steps of the recipe (which also include ingredients) to determine how similar each recipe is.

For a recipe having a combination of common ingredients (e.g "Salt") and uncommon ingredients (e.g "Bacon"), more importance can be given to the uncommon ingredients. A widely adopted method for addressing this problem involves employing the Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer. TF-IDF is a numerical metric designed to indicate the significance of a word within a document collection or corpus. TF represents the frequency of a word in a specific document, while IDF is the inverse of the document frequency across the entire document corpus. This metric aids in capturing the essential ingredients of each recipe by assigning higher weights to less frequently occurring ingredients. The formula for calculating the weight (w_{ij}) for an item is expressed as

$$w_{i,j} = tf_{i,j} \log\left(\frac{N}{df_i}\right)$$

where $tf_{i,j}$ is the frequency of term i in document j, df_i is the number of documents containing term i, and N is the total number of documents in the corpus.

Subsequently, a vector space model is employed to assess the similarity between the document vectors. In this model, each item is represented as a vector of its attributes (which are also vectors) within an n-dimensional space. The similarity between two vectors is determined by calculating the angle between them and calculating the cosine similarity. Cosine similarity is calculated between the given recipe and the recommended recipe. Here, we take the cosine between two vectors representing those recipes. Mathematically,

$$cosine(x, y) = \sum_{i=1}^{n} \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}}$$

In matrix form, cosine similarity between recipes (users) can be expressed as:

$$S = \frac{M_A * M_B^T}{||M_A|| * ||M_B||}$$

$\boldsymbol{M_A}$ and $\boldsymbol{M_B}$ are the normalised user-item matrices for different recipes (users), $||\boldsymbol{M_A}||$ and $||\boldsymbol{M_B}||$ are the Euclidean norms of the matrices and **S** is the cosine similarity matrix.

Content-based filtering does not rely on the data contributed by other users to make recommendations to a single user. It is noteworthy that the content-based approach tends to recommend recipes aligning with the established taste preferences of an individual, without introducing new recipe suggestions unless explicitly mentioned by the user. This limitation can be solved through collaborative filtering, an approach detailed below.

5.3. COLLABORATIVE FILTERING

Collaborative filtering involves making recommendations based on the preferences and behaviours of multiple users. In item-to-item collaborative filtering, recommendations are generated by identifying items that are such as those the user has liked or interacted with. On the other hand, user-to-user collaborative filtering recommends items based on the preferences of users who are similar to the target user. These methods leverage a similarity matrix to identify patterns and similarities between users or items, aiming to enhance the efficiency and comprehensiveness of recommendation models. Unlike content-based and knowledge-based approaches, collaborative filtering considers the collective preferences of multiple users, making it a more inclusive and potentially more effective recommendation strategy.

5.3.1. ITEM-ITEM COLLABORATIVE FILTERING

A basic collaborative filtering model looks at how users rate recipes. For example, if two recipes, say R1 and R2, have similar ratings from users, the model suggests one when the user likes the other. Imagine there are only two users, U1 and U2, who rated three recipes - R1, R2, and R3. If R1 and R2 have close ratings on a chart, the model recommends R2 to someone who liked R1, and vice versa. It finds the closest recipes and suggests them based on how much they resemble each other.

5.3.2. USER-USER COLLABORATIVE FILTERING

In this method, recipe recommendations depend on deciding how similar two users' tastes are. This approach involves accessing the recipe preferences of a user, denoted as U1. By evaluating the recipes favoured by U1 and cross-referencing them with those liked by other users who share similar preferences, the method aims to establish correlations. Subsequently, it suggests the recipes that are tried by the similar users but not by U1. The recommendation process entails identifying the recipe with the highest correlation in this subset and suggesting it to user U1.

5.3.2.1. Neighbourhood Model

In these neighbourhood-based approaches, the similarity between users can be calculated in three ways, using Euclidean Distance, Cosine Similarity or Pearson Correlation Coefficient. Four different variations of the KNN algorithm is considered – KKNNBaseline, NNBasic, KNNMeans and KNNZScore.

KNN Basic (uses only similarity matrix)	$r_{u,m} = \frac{\sum_{v \in N(u)} sim(u,v) r_{v,m}}{\sum_{v \in N(u)} sim(u,v)}$
KNN Baseline (also uses baseline rating)	$r_{u,m} = b_{u,m} + \frac{\sum_{v \in N(u)} sim(u,v)(r_{v,m} - b_{vm})}{\sum_{v \in N(u)} sim(u,v)}$
KNN Means (also uses mean ratings of each user)	$r_{u,m} = \mu_u + \frac{\sum_{v \in N(u)} sim(u,v)(r_{v,m} - \mu_v)}{\sum_{v \in N(u)} sim(u,v)}$
KNN Z-Scores (also uses Z-scores normalisation of each user)	$r_{u,m} = \mu_u + \sigma_u \left\{ \frac{\sum_{v \in N(u)} sim(u,v) \left(\frac{r_{v,m} - \mu_v}{\sigma_v} \right)}{\sum_{v \in N(u)} sim(u,v)} \right\}$

5.3.2.2. Matrix Factorization

Since not all recipes have been rated and not all users have given a rating, the matrix of users vs recipe is very sparse. This method translates the user-item matrix into a lower-dimensional space to facilitate the prediction. Through matrix decomposition, we can break down a larger matrix into its constituent parts making the calculations simpler and more efficient. To predict the missing values of the matrix, we find the dot product between the **P** "similarities of user with the features" and the **Q** "similarities of the recipes with the features". We obtain R matrix, which is expressed as:

$$R \cong PxQ^T = \hat{R}$$

Depending on **R** matrix formulation, we can have two matrix factorization models – SVD and SVDpp.

5.3.2.3. SVD

This method is equivalent to probabilistic matrix factorization and can be mathematically formulated as explained below. Under this method, the prediction $\hat{r}_{ui}$ is set as:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

If user u is unknown, then the bias b_u and the factors p_u are assumed to be zero. The same applies for item I and b_i and q_i.

To estimate all the unknown, we minimize the following regularized squared error:

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - \hat{r}_{ui})^2 + \lambda \left(b_i^2 + b_u^2 + \left| |q_i| \right|^2 + \left| |p_u| \right|^2 \right)$$

The minimization is performed by a very straightforward stochastic gradient descent:

$$b_u < - b_u + \gamma(e_{ui} - \lambda b_u);\;\; b_i < - b_i + \gamma(e_{ui} - \lambda b_i);$$

$$p_u < - p_u + \gamma(e_{ui} q_i - \lambda p_u);\;\; q_i < - q_i + \gamma(e_{ui} p_u - \lambda q_i)$$

where $e_{ui} = r_{ui} - \hat{r}_{ui}$. These steps are performed over all the ratings of the trainset and repeated n_epochs times. Baselines are initialized to 0. User and item factors are randomly initialized according to a normal distribution, which can be tuned using the init_mean and init_std_dev parameters.

You can choose to use an unbiased version of this algorithm, simply predicting: $\hat{r}_{ui} = q_i^T p_u$

5.3.2.4. SVDpp

This is an extension to SVD and takes into account the implicit ratings. The predictions $\hat{r}_{ui}$ is set as:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right)$$

5.4. MODEL COMPARISON

Table 2: Strengths and Weaknesses

Model	Strengths	Weaknesses
Knowledge-Based Filtering	• Highly customizable • Ideal when large dataset available. • Can handle the cold start problem, i.e., it can recommend recipes to a new user.	• Cannot recommend diverse recipes. • Requires maintaining a frequently updated database with recent user preferences for better recommendations.

Content-Base Filtering	• Highly personalised based on user's preferences. • Easier to explain to users as recipes are recommended based on features of the recipe (here ingredients).	• Cannot recommend diverse recipes, due to lack of user history. • Recommendation dependent on the feature of recipe chosen to assess similarity. • Cannot be an adaptable model with changing user preferences.
Collaborative Filtering	• Can recommend diverse recipes, based on user preferences. • Detailed dataset for recipes is not required and user-recipe interactions play a major role.	• Difficult to process model when user-recipe interactions are less. • Not possible to recommend for new users with no user history. • Computationally intense. • Hard to include side features of recipes

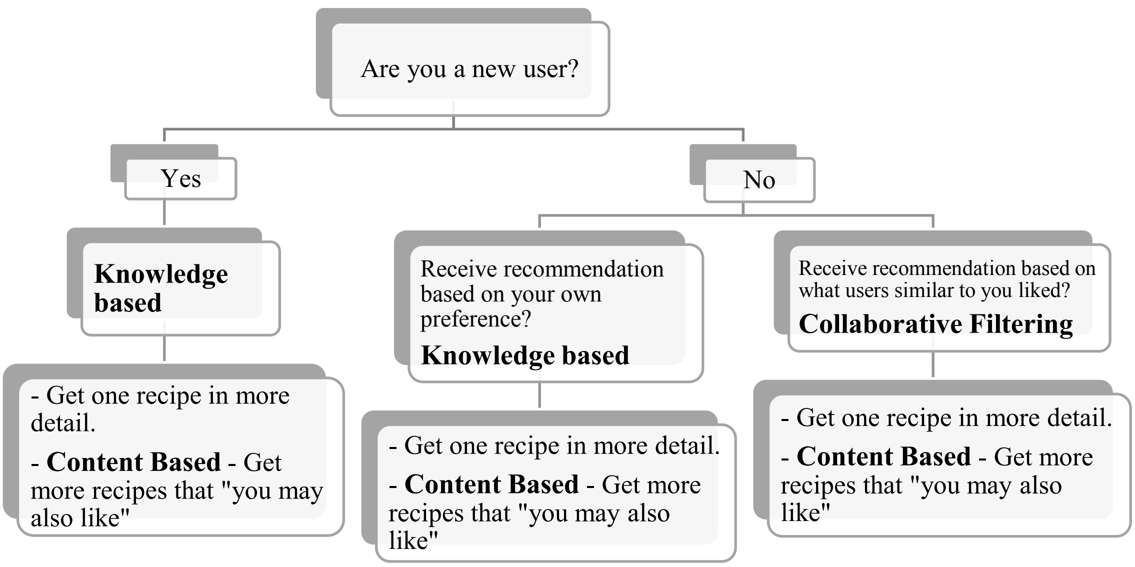
5.5. HYBRID MODEL

As discussed earlier, each filtering strategy possesses certain advantages and limitations. In pursuit of constructing an enhanced recommendation system, we propose the development of a hybrid model that integrates multiple models using the switching approach. The sequence of employing these models is contingent upon several factors, including the availability of the user's historical data, the status of the user (whether new or existing), and the user's preferences concerning the nature of recipe recommendations.

The hybrid model adopts a flexible approach, considering the varying circumstances surrounding user interactions. For users with an established history, a personalised recommendation can be initiated based on collaborative filtering methods such as User-User or Item-Item Collaborative Filtering. Further refinement of recommendations is achieved through the incorporation of matrix factorization techniques like Singular Value Decomposition (SVD) and its extensions, such as SVD++. Conversely, for new users lacking historical data, knowledge-based filtering may serve as an initial recommendation strategy. Extra fine tuning can be done using content-based filtering to help the user find more recipes similar to what they like.

The decision on the sequencing of these models is dynamic and considers additional user preferences, such as whether the user desires recommendations similar to a particular recipe or prefers suggestions based on the preferences of users with similar tastes. This adaptability underscores the hybrid model's versatility in accommodating diverse user scenarios, thereby striving to provide more nuanced and effective recipe recommendations.

Chart 1: Hybrid Model



6. NUMERIC EXPERIMENTS

6.1. COMPARISON OF COLLABORATIVE FILTERING MODELS

We will test different Matrix Factorization-based algorithms and k-NN inspired algorithms to determine which has the lowest RMSE and which has the lowest MAE, when using the default parameters.

Table 3: Model performance under default parameters

Algorithm	test_rmse	test_mae	fit_time	test_time
SVDpp	**0.561213**	**0.357518**	4.014064	3.009271
SVD	0.565027	0.364604	0.848474	0.209999
KNNBaseline	0.592084	0.372148	0.970071	2.485883
KNNWithMeans	0.597708	0.369052	0.916874	2.209857
KNNWithZScore	0.610445	0.366837	0.991537	2.201809
KNNBasic	0.632862	0.392884	0.776	2.002055

Tuning the hyperparameters of each model and testing which has the lowest RMSE:

Table 4: Tuned hyperparameters for each model

Algorithm	**Best_Params**	**RMSE**
SVDpp	{'n_factors': 50, 'n_epochs': 20, 'lr_all': 0.005, 'reg_all': 0.2}	0.558956
SVD	{'n_factors': 50, 'n_epochs': 20, 'lr_all': 0.005, 'reg_all': 0.2}	0.559563
KNNBaseline	{'k': 40, 'sim_options': {'name': 'cosine', 'user_based': True}, 'bsl_options': {'method': 'sgd'}}	0.58812
KNNWithMeans	{'k': 40, 'sim_options': {'name': 'cosine', 'user_based': True}}	0.596017
KNNWithZScore	{'k': 40, 'sim_options': {'name': 'cosine', 'user_based': False}}	0.601471
KNNBasic	{'k': 40, 'sim_options': {'name': 'cosine', 'user_based': False}}	0.607869

SVDpp has the lowest RMSE. However, it only performs slightly better than SVD, despite having a much greater time complexity. We will therefore use the SVD model with the following hyperparameters: 'n_factors': 50, 'n_epochs': 20, 'lr_all': 0.005, 'reg_all': 0.2.

On comparing the SVDpp, SVD, KNNBasline, KNNMeans, KNNZScore and KNN Basic, we have found that SVDpp has the lowest RMSE and MAE, with the default hyperparameters. On tuning the hyperparameters, we see that SVDpp performs the best, while SVD is second with a very slight increase in RMSE. Since SVDpp has very high time complexity, as compared to SVD, we choose SVD as the better model for recipe recommendation. Finally, we choose SVD model in our switching hybrid model.

6.2. SWITCHING HYBRID MODEL

Knowledge-Based Filtering

The recipe recommendations from knowledge-based filtering model obtained for a specific user preference is explained below. For the user inputs tabulated below.

Do you have a preference on ingredients? (yes/no)	*Yes*
Please enter preferred ingredients separated by commas (e.g., ingredient1,ingredient2)	*chicken, bacon*
Should the recipes include all of these ingredients? (yes/no)	*Yes*
Are you limited for time? (yes/no)	*Yes*
How long do you have in minutes?	*45*
Would you like to limit calorie intake? (yes/no)	*No*
Would you prefer a recipe with (1. fewer steps, 2. fewer ingredients, 3. less time required, 4. the highest rating, 5. lower calories, 6. higher calories, 7. specific nutritional preferences?) Please select from options 1-7 (enter the number)	*7*
Enter your specific nutritional preferences, separated by commas (e.g. sodium (low), protein (high)). You can choose from: (total fat, saturated fat, sugar, sodium, protein, carbohydrates)	*protein (high), sodium (low)*
What is the maximum number of recommendations you'd like to receive? Enter a numerical value	*5*

The recipe recommendations are.

	name	recipe_id	minutes	steps	ingredients	calories	total fat (PDV)	sugar (PDV)	sodium (PDV)	protein (PDV)	saturated fat (PDV)	carbohydrates (PDV)
138127	sweet n spicy bacon chicken	333452	30	['preheat oven to 350 degrees fahrenheit', 'slice chicken into bite-sized pieces', 'cut bacon into 1 to 1-1 / 2 inch pieces', 'wrap a piece of bacon around each chicken piece', 'use toothpicks if necessary', '...	['chicken', 'bacon', 'brown sugar', 'cayenne pepper']	609.4	65.0	106.0	23.0	55.0	67.0	9.0
16904	blt barbecue chicken salad	459616	10	['in a small bowl , mix the mayo , bbq sauce , onion , lemon juice , and pepper', 'cover & refrigerate until ready to serve', 'place salad greens on a large serving platter , add tomatoes , chicken , bacon and...	['mayonnaise', 'barbecue sauce', 'onions', 'lemon juice', 'pepper', 'chicken', 'bacon', 'hard-boiled eggs', 'salad greens', 'tomatoes']	331.6	36.0	17.0	13.0	39.0	31.0	3.0
28747	chicken and bacon pan fried sandwich	434019	26	['place the bacon under a preheated hot grill and cook for 1-2 minutes or until crispy', 'place the maonnaise , garlic , lemon juice , chives and chicken in a bowl and mix well to combine', 'spread both sides ...	['bacon', 'mayonnaise', 'garlic cloves', 'lemon juice', 'chives', 'chicken', 'butter', 'sourdough bread', 'smoked cheddar cheese']	673.0	43.0	15.0	47.0	61.0	74.0	24.0
104780	pear blue cheese walnut and bacon salad	476046	23	['whisk together all vinaigrette ingredients until well blended', 'set aside', 'in a small frying pan set on a medium heat , melt the butter , stir in the maple syrup or honey and then stir in the walnuts', 'c...	['olive oil', 'limes', 'white wine vinegar', 'dijon mustard', 'maple syrup', 'green onions', 'salt and pepper', 'walnut halves', 'butter', 'cayenne', 'arugula', 'bartlett pears', 'blue cheese', 'bacon', 'chick...	354.6	44.0	43.0	13.0	17.0	33.0	6.0
121680	sauteed swiss chard with bacon	362648	20	['coat a large saute pan lightly with olive oil and add the diced bacon , garlic , and crushed red pepper', 'bring the pan to medium-high heat', 'when the garlic has turned a lovely golden brown , remove from ...	['olive oil', 'bacon', 'garlic cloves', 'crushed red pepper flakes', 'swiss chard', 'chicken', 'kosher salt']	20.5	0.0	4.0	8.0	3.0	0.0	1.0

Would you like to (1. see a recipe in more detail, 2. see recipes similar to one of the recommendations, 3. Quit) Please select option 1, 2 or 3	2

Content-Based Filtering

Enter the recipe ID to see similar recipes	476046

	name	recipe_id	minutes	steps	ingredients	calories	total fat (PDV)	sugar (PDV)	sodium (PDV)	protein (PDV)	saturated fat (PDV)	carbohydrates (PDV)
109990	potato salad with creamy blue cheese dressing	263131	20	['peel potatoes if you wish , then cut into bite-size chunks', 'place in a large saucepan of salted water , set over high heat and bring to the boil', 'pa...	['potatoes', 'lemon juice', 'bacon', 'red pepper', 'green onions', 'mayonnaise', 'sour cream', 'hot pepper sauce', 'blue cheese', 'salt and pepper']	339.6	26.0	20.0	18.0	17.0	33.0	13.0
132531	sweet potato pear soup	115291	85	['melt the butter in a large pot over a medium heat , and saut onion for 2-3 minutes until softened but not brown', 'add the diced sweet potato and diced ...	['butter', 'yellow onions', 'sweet potatoes', 'pears', 'chicken broth', 'white wine', 'greek yogurt', 'of fresh mint']	223.7	5.0	70.0	26.0	12.0	8.0	14.0
88173	maple glazed walnuts	253308	15	['preheat a dry skillet over a medium-high heat', 'add the walnuts , maple syrup and salt', 'cook , stirring frequently , until syrup is caramelized and n...	['walnut halves', 'maple syrup', 'salt']	301.7	39.0	46.0	2.0	11.0	12.0	5.0
85536	loaded baked potato chicken casserole	505862	90	['preheat oven to 500 degrees', 'in large bowl mix olive oil , salt , pepper , paprika , garlic powder and hot sauce', 'add potatoes to bowl and allow to ...	['chicken breasts', 'potatoes', 'olive oil', 'salt', 'fresh ground pepper', 'paprika', 'garlic powder', 'hot sauce', 'cheddar cheese', 'bacon', 'green oni...	652.2	57.0	9.0	41.0	75.0	65.0	13.0
7913	bacon wrapped smoked gouda stuffed chicken breasts	111169	30	['combine , cayenne , garlic powder , paprika and pepper', 'flatten chicken breasts to approx 1 / 4" thickness', 'season both sides of chicken with the ca...	['boneless chicken breasts', 'bacon', 'smoked gouda cheese', 'cayenne', 'garlic powder', 'paprika', 'pepper']	365.0	35.0	2.0	13.0	73.0	44.0	0.0

Collaborative Filtering

Are you a new user? (yes/no)	*No*
Welcome back user 768828!	
Would you like recommendations based on (1. your preferences, 2. what users similar to you liked) Enter the corresponding number (1/2)	*2*
What is the maximum number of recommendations you'd like to receive? Enter a numerical value	5

	name	recipe_id	minutes	steps	ingredients	calories	total fat (PDV)	sugar (PDV)	sodium (PDV)	protein (PDV)	saturated fat (PDV)	carbohydrates (PDV)
14597	benihana ginger salad dressing	1985	10	['combine all ingredients in a blender', 'blend on high speed for about 30 seconds or until all of the ginger is well pureed']	['onion', 'peanut oil', 'rice vinegar', 'water', 'fresh ginger', 'celery', 'ketchup', 'soy sauce', 'sugar', 'lemon juice', 'garlic', 'salt', 'pepper']	90.0	13.0	6.0	9.0	0.0	7.0	0.0
21961	brownies in a jar	1904	10	['pour sugar into a clean and dry one quart jar', 'press down firmly', 'add cocoa powder and press down firmly', 'pour in chopped pecans , making sure they are evenly layered', 'combine flo...	['white sugar', 'cocoa', 'pecans', 'all-purpose flour', 'baking powder', 'salt', 'butter', 'eggs']	183.8	13.0	75.0	7.0	4.0	20.0	8.0
28540	chicken and black bean enchiladas	1522	35	['julienne chicken', 'saute bacon until crisp , reserving 2 tbsp drippings', 'saute chicken and garlic in drippings until chicken is opaque', 'add 1 / 2 cup picante sauce , beans , red bell...	['boneless skinless chicken breast half', 'bacon', 'garlic cloves', 'picante sauce', 'red bell pepper', 'cumin', 'salt', 'green onion', 'monterey jack cheese', 'black beans', 'flour tortill...	645.2	32.0	21.0	61.0	70.0	42.0	26.0
36812	classic cream scones	192	85	['preheat oven to 425f', 'lightly butter a baking sheet', 'in a large bowl , stir together the flour , sugar , baking powder , and salt', 'cut the butter into 1 / 2-inch cubes and distribut...	['all-purpose flour', 'granulated sugar', 'baking powder', 'salt', 'unsalted butter', 'heavy cream', 'egg', 'vanilla extract', 'currants']	168.2	12.0	28.0	3.0	5.0	24.0	7.0
45749	dark chocolate cake	2496	60	['heat oven to 350f', 'grease and flour two 9 inch round baking pans or one 13x9 inch pan', 'in large mixer bowl , stir together dry ingredients', 'add eggs , milk , oil , and vanilla', 'be...	['sugar', 'flour', 'baking cocoa', 'baking powder', 'baking soda', 'salt', 'eggs', 'milk', 'vegetable oil', 'vanilla extract', 'boiling water']	189.1	10.0	80.0	10.0	5.0	7.0	10.0

7. CONCLUSIONS

In conclusion, the evaluation of a recipe recommendation system represents a multifaceted approach. By implementing collaborative, content-based and knowledge-based filtering in a switching hybrid model, our research optimises the precision and relevance of recipe suggestions. Numerical experiments, including algorithm performance evaluations and hyperparameter tuning, have been instrumental in refining the recommendation algorithms. Addressing the cold start problem and temporal dynamics, by the logical arrangement of models in the hybrid model, has underscored the adaptability of the recommendation system. The system's ability to evolve with changing user behaviours and preferences positions it as a valuable tool in the culinary domain.

Looking ahead, continuous refinement and enhancement of the recommendation algorithms, informed by user feedback and emerging trends, will be pivotal in maintaining the system's relevance and utility. Our research contributes to the broader discourse on recommendation systems, demonstrating the applicability of traditional filtering techniques in the realm of culinary exploration.

8. LIMITATIONS

(i) Data limitations: The dataset used for recipe recommendations in this study is very large. Therefore, the execution of content-based and collaborative filtering models had high time complexity. This computationally challenging task was performed by filtering the dataset further by applying certain logical criterions (e.g. excluding recipes with fewer than 20 ratings), to enhance system efficiency.

(ii) Model limitations: Under our hybrid model, we tried to combine collaborative filtering with knowledge-based filtering. This combination would enable recommending recipes based on what similar users liked and filtering them by specific ingredients the user wants. However, we were not able to combine these two models due to computational inefficiency.

9. SCOPE OF STUDY

(i) Real-Time Data Integration: Enhancing the efficiency of recommendation system models by the incorporation of real-time data concerning users' preferences and choices. The acquisition of such data could be achieved through dynamic sources, including web scraping from platforms such as Deliveroo and UberEats. This integration aims to augment the personalization and relevance of recipe recommendations for individual users.

(ii) Reinforcement Learning Utilising User Feedback: Models can be further improved through the systematic accumulation of user feedback regarding recommendations provided by each model. The incorporation of user feedback allows the model to iteratively refine its understanding of user preferences. Employing reinforcement learning, the model can dynamically adjust its recommendation strategy by assimilating user feedback on recommended recipes, thereby contributing to the development of an improved user profile for future interactions.

10. REFERENCES

- Recipe and Interactions Dataset (Kaggle, 2014) https://www.kaggle.com/datasets/shuyangli94/food-com-recipes-and-user-interactions?select=RAW_recipes.csv
- A systematic review & research perspective on recommender systems by Deepjyoti Roy & Mala Dutta (2022) https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00592-5
- Recommender System — Matrix Factorization by Denise Chen (2020) https://towardsdatascience.com/recommendation-system-matrix-factorization-d61978660b4b
- Content-based filtering and Collaborative filtering by Nikola Kuzmic (2021) https://towardsdatascience.com/user-to-user-vs-item-to-item-collaborative-filtering-explained-through-simple-examples-part-1-f133bec23a58
- Cosine Similarity Matrix by Zuzanna Deutschman (2023) https://neptune.ai/blog/recommender-systems-metrics
- Collaborative Filtering (2015) https://surprise.readthedocs.io/en/stable/prediction_algorithms_package.html
- Knowledge-based Recommender by Michael Fuchs (2020) https://michael-fuchs-python.netlify.app/2020/10/01/recommendation-systems-knowledged-based-recommender/