

# Recipe Recommender System

1. Create a virtual environment
2. pip install -r requirements.txt
3. python run main.py

Github Repo : <https://github.com/SushOS/Recipe-Recommender>

## Overview

**Dataset used :** [Food.com Recipes and User Interactions](#)

This project implements a recipe recommendation system in which user can dynamically enter their preferences in a **terminal-based** UI, and the system **filters** + **ranks** recipes accordingly. If no recipes match, a **fallback** mechanism offers the "closest possible" matches.

- Builds a **content-based recommendation** approach using **TF-IDF**.
- **Incorporates a collaborative filtering** component using average ratings for recipes.
- Users can specify **dietary restrictions** (e.g., vegetarian, non\_vegetarian, vegan, gluten\_free, dairy\_free), **cuisine preference**, **available ingredients**, **time**, **difficulty**, etc.
- **Collects** user feedback and applies **NLP** on textual reviews to **infer** additional preferences (e.g., "concerned about calories").
- **Visualizes** the dataset and recommendations via **Seaborn/Matplotlib**.

## Project Structure

```
recipe_recommender/  
|--- data/
```

```

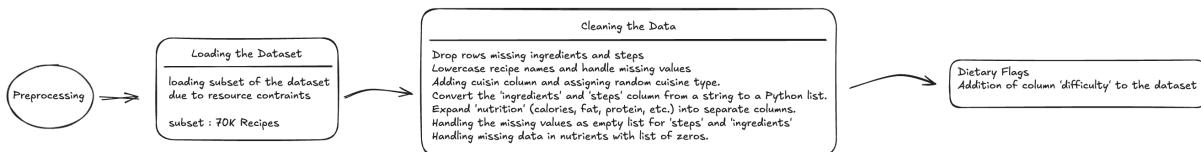
|   └── feedback.json          # JSON file storing user fee
dback
|   └── raw/
|       ├── RAW_recipes.csv    # The main recipes dataset
|       ├── RAW_interactions.csv # The user interactions/rati
ngs data
└── src/
    ├── main.py
    ├── data_preprocessing.py
    ├── recommendation_engine.py
    ├── feedback_manager.py
    ├── nlp_preference_extraction.py
    ├── utils.py
    └── visualization.py
└── requirements.txt

```

## What are each of the files doing?

1. main.py : The entry point. Prompts user for preferences, runs the pipeline, displays top recommendations, collects feedback.
2. data\_preprocessing.py : Cleans the dataset, randomly assigns cuisines, handles nutrition info, and creates TF-IDF embeddings.
3. recommendation\_engine.py : Core recommendation logic (content-based + CF), plus fallback filtering.
4. feedback\_manager.py : Loads/saves user feedback in JSON.
5. nlp\_preference\_extraction.py : Simple keyword-based approach for scanning user comments (NLP).
6. utils.py : Common helper functions (validate preferences, format recipe output).
7. visualization.py : Functions for plotting distributions, correlation, top recommendations, etc.

# Preprocessing Steps



**Difficulty calculation :**  $(\text{number of ingredients})/10 + (\text{cooking\_time})/60 + (\text{n\_steps})/5$

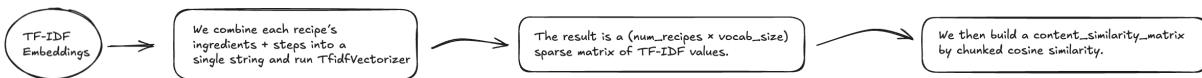
## Dietary Flags:

- For 'vegetarian', 'vegan', 'gluten\_free', 'dairy\_free', we store large lists of "excluded" items (like 'beef', 'chicken', etc.) and check if each recipe mentions them.
- Mark each recipe as `is_vegetarian`, `is_vegan`, etc. if it **does not** contain any of those terms.
- Also create `is_non_vegetarian` = `~is_vegetarian` for those wanting strictly non-veg.

# TF-IDF Content Embeddings

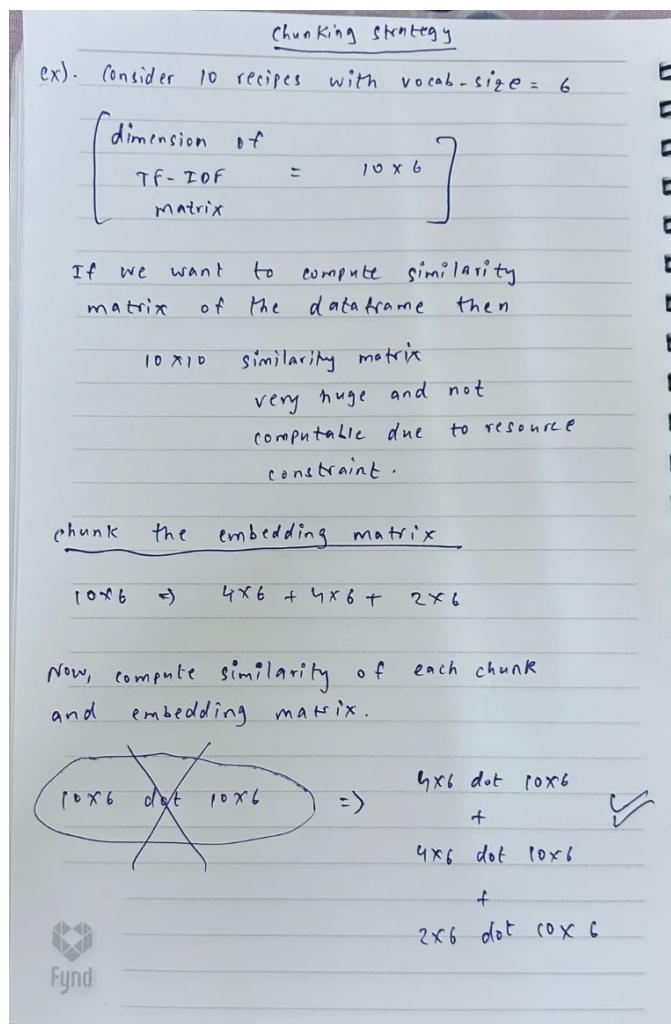
## What is TF-IDF?

- **Term Frequency (TF):** How often a word appears in a recipe's text relative to the total words in that recipe. A recipe mentioning "chicken" 5 times out of 50 total words has a higher TF for "chicken" than one mentioning it once out of 100 words.
- **Inverse Document Frequency (IDF):** Gives **less** weight to words that appear **very frequently** across all recipes (like "salt", "bake") and **more** weight to rarer, more distinguishing words (like "tamarind" or "mozzarella")



## Chunking Strategy

The below image shows how chunking strategy is used to compute the similarity matrix between the recipes.



## Content Based Score Computation

Content-Based Scoring

ingredient weight = 5 ( $w_1$ )  
 nutrient weight = 3 ( $w_2$ )  
 time weight = 1 ( $w_3$ )

Score = 0

1). Ingredient component

$\text{score} \leftarrow \frac{\text{number of overlap of user ingredient and ingredient list}}{\text{len(ingredient list)}} \times w_1$

2). Nutrient component

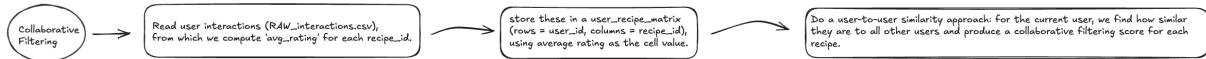
$\text{diff} = \text{abs}(\text{cur-val} - \text{target})$   
 $\text{max-val} = \max(\text{df}[\text{nutrient}])$   
 $\text{inv-diff} = \text{max-val} - \text{diff}$   
 $\text{score} \leftarrow \text{inv-diff} \times w_2$   
 ↳ for all nutrients

3). cooking time

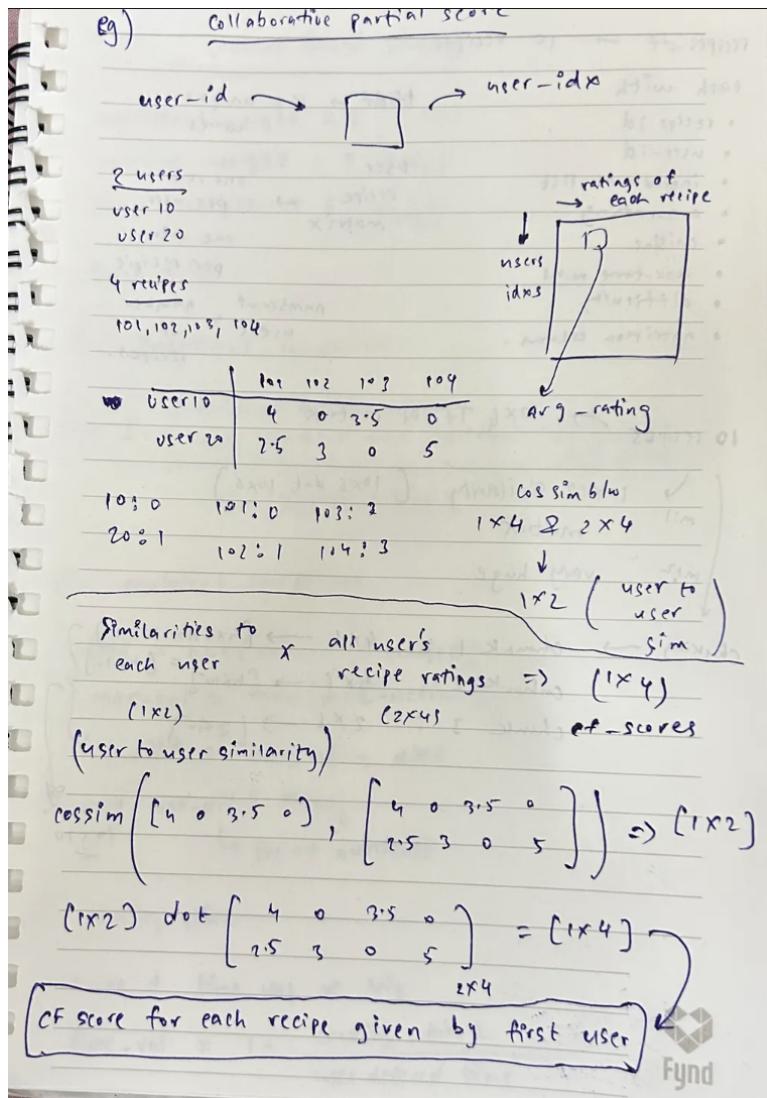
$\text{score} \leftarrow \text{time-val} \times w_3$   
 $\text{time-val} = 1 - \frac{\text{cooking-time}}{\text{user defined time}}$

Fynd

## Collaborative Filtering Score Computation



The below image shows the collaborative scores of each recipe due to a particular user. Similarly the collaborative scores of the recipes due to other users also add up based on the user to user similarity.



## Blending Scores + Filters

1. **Content-based** partial score: Weighted sum of ingredient overlap, nutrition closeness, and time-based factor.
2. **Collaborative** partial score: If the user is recognized, we find a user-based similarity score.
3. **Final** =  $0.7 \times \text{content} + 0.3 \times \text{CF}$ . (We can change the weights as per our needs)
4. **Filter** out recipes not matching cuisine, dietary restrictions, time, difficulty, must-have ingredients.

5. If no match remains, we do a **fallback**: skip cuisine/dietary checks but keep time/difficulty/ingredients.

## Collecting & Parsing User Feedback (NLP)

**Feedback** is stored in a JSON file (feedback.json) with entries like:

```
{  
    "123": [  
        { "recipe_id": 26067, "rating":2.0, "comments":"Has too  
        many calories than I need." },  
        ...  
    ]  
}
```

- **NLP**: We have nlp\_preference\_extraction.py scanning each user's comment for keywords: e.g., "calorie", "healthy."
- If user mentions "calorie," we interpret that they're concerned about **low\_calorie**.
- We update user\_preferences['nutrition\_preferences']['calories'] to something smaller (e.g. 400) automatically.
- This ensures **textual** feedback influences the final recommended recipes.

## Visualization

We included several plotting functions in visualization.py:

1. **plot\_cuisine\_distribution**: Bar chart of how many recipes per cuisine.
2. **plot\_top\_recommendations**: Bar chart of recommended recipes by their average rating.
3. **show\_recommendations\_cuisine\_breakdown**: Pie chart of recommended recipes by cuisine.

4. **plot\_difficulty\_distribution**: Histogram of difficulty from 1–5.
5. **plot\_rating\_distribution**: Distribution of avg\_rating.
6. **plot\_diet\_flags\_breakdown**: How many recipes meet is\_vegetarian, is\_vegan, etc.
7. **plot\_correlation\_matrix**: A heatmap correlation among numeric columns (e.g., avg\_rating, difficulty, calories, etc.) ignoring cook\_time\_minutes.

## Why This Approach?

Below is explanation of **why** we chose this **hybrid (content-based + collaborative filtering)** approach, **why it's a good fit** for this particular dataset, and **how it compares** to other possible methods.

### 1. Rich Textual Data (Ingredients & Steps)

- The dataset includes **detailed textual descriptions** (ingredients, cooking steps). A **content-based** method leveraging **TF-IDF** captures how recipes differ in terms of keywords (e.g., "tomato," "cumin," "frying," "baking"). This is **ideal** for recommending recipes that share similar **ingredient or step** patterns.

### 2. User Ratings & Reviews

- The dataset also has **ratings** (in RAW\_interactions.csv), which reflect **user preferences**. A **collaborative filtering (CF)** module uses these ratings to push recipes that **similar users** enjoyed. This adds **personalization** on top of purely content-based matching.

A **hybrid system** of TF-IDF + average-rating CF is **straightforward** to implement and interpret.

TF-IDF embeddings and chunk-based cosine similarity are well-supported by scikit-learn. The CF step is also manageable if we keep a user-recipe matrix for average ratings which we have done. We can handle moderate data sizes without needing a huge GPU cluster.

### 3. Fallback & Filtering

- Because we can derive **scores** from both content and CF, we can also **filter** out recipes by time constraints, dietary needs, or must-have ingredients, and

still produce a final ranking. Many other methods (e.g., deep neural models) may not handle these constraints as transparently or easily.

## Why Hybrid (TF-IDF + CF) Is Beneficial

### 1. Handles Both Cold Start and Personalization

- The content-based side helps recommend new or rarely rated recipes because they share textual similarity with known recipes.
- The CF side ensures personalization, surfacing recipes that **similar users** enjoyed.
- You can **explain** recommendations: "This dish is recommended because it shares ingredients with your favourite dish and has a decent average rating from similar users."

### 2. Flexible & Expandable

- Easy to **add** new constraints (time, dietary flags) or new rating data.
- If the dataset or user base grows, you can incorporate more advanced CF or deeper textual embeddings while still leveraging the same framework.

### 3. Balance of Complexity & Performance

- It's not overly complicated to implement or tune, yet it typically performs **much better** than a single method alone (pure content or pure CF).
- Tweaking and weighting between content-based and CF for each domain or user scenario.

## Challenges Faced

1. **Data Inconsistency:** Some recipes have missing ingredients or steps. Filled them with empty instances or zero values.
2. **Random Cuisine Assignment:** The dataset didn't have a "cuisine" column, so we randomly assigned it. That's not truly reflective of real data, but we did so to show that the model is capable to filter based on the cuisines.

3. **Dietary Patterns:** The list of forbidden items for “vegetarian/vegan” can be endless. Typos or brand names might slip through. Tried to incorporate all the keywords.
4. **NLP** implemented is very **basic** (keyword-based). Real user comments may have more nuance.

## Potential Improvements

1. **Advanced NLP:** Use spaCy or a large language model to detect user preferences more accurately (sentiment, synonyms, brand names).
2. **Better Cuisine Labelling:** Instead of random assignment, classify recipes by actual ingredients (e.g., “soy sauce,” “rice vinegar” → “Chinese”).
3. **Matrix Factorization:** Replace the simple average rating user→user approach with an advanced SVD or neural CF method for better personalization.
4. **User Interface:** Create a **web app** or **Streamlit** dashboard so users can select preferences and see real-time updates.
5. **Scalability:** For extremely large datasets, consider approximate nearest neighbour methods for TF-IDF similarity, or store data in a specialized index.

## Conclusion

This project **blends** a **content-based** approach (TF-IDF of ingredients/steps) with a **collaborative filtering** approach (user rating data) and respects **user constraints** (time, difficulty, dietary flags). We incorporate **NLP** on textual reviews to infer hidden preferences like “concerned about calories.” The system is demonstrated through a **terminal-based** user flow, plus a suite of **visualizations** that help explore both the dataset and the final recommendations.

Overall, it serves as a foundation for a **hybrid** recipe recommender, with ample room for **further expansions** and **enhancements** in modeling, UI, and data integration.

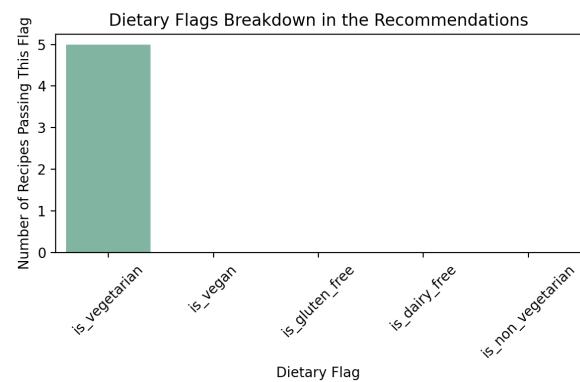
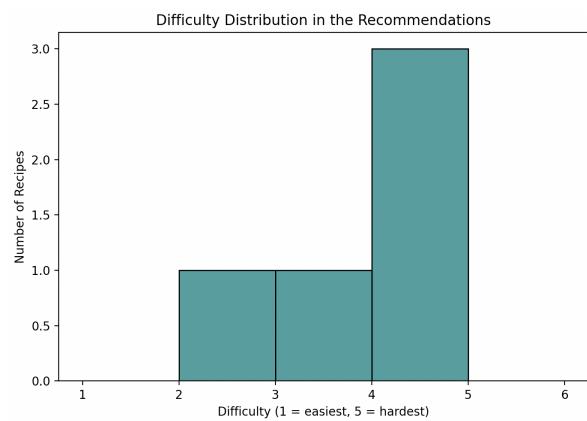
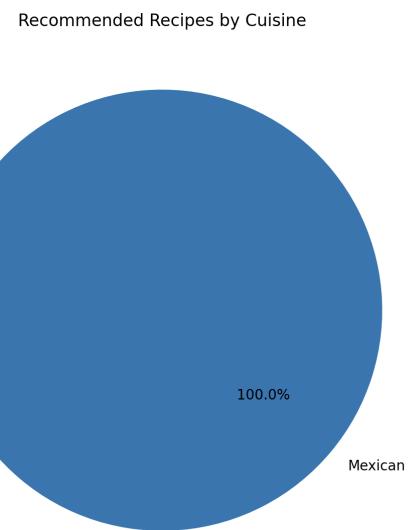
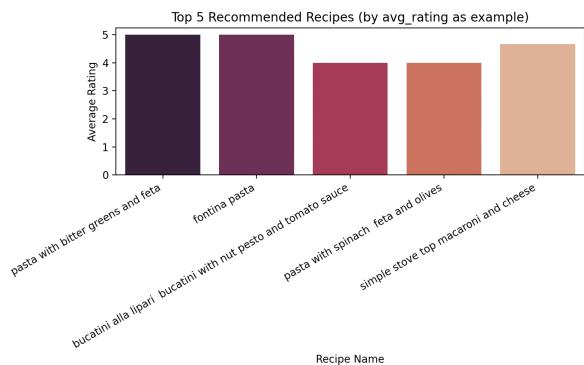
# Dataset Visualizations



## 1. User Preference

```
3. Please enter your user preferences.
Enter a user ID (integer): 5849
Enter cuisine preference (e.g., 'Indian', 'Italian', 'Mexican', 'Chinese', American'): Mexican
Enter dietary restrictions (comma-separated, e.g. 'vegetarian, non_vegetarian, gluten_free', 'vegan'): vegetarian
Enter available time in minutes (integer): 50
Enter available ingredients (comma-separated, e.g. 'tomatoes, onions, etc..'): pasta
Enter max difficulty (1-5, float allowed): 5
Enter target calories (float): 1500
Enter target protein in grams (float): 10
Enter target fat in grams (float): 10
```

## Visualizations



**The top 5 recommendations and modified user preference using NLP based on the comments given to the recipe by the user**

5. Top Recommendations:

Recipe 1:  
name: pasta with bitter greens and feta  
cook\_time: 20 minutes  
ingredients: ['olive oil', 'yellow onions', 'greens', 'salt', 'pasta', 'feta cheese', 'parmesan cheese', 'fresh ground black pepper']  
instructions: heat the oil in a deep skillet | add the onions and saute until soft | add the greens and cook for 2-3 minutes over medium heat | while the onions are cooking , start the pasta water on high heat in a separate large pot | when the onions are soft, add the feta to the skillet | add salt and stir until the greens start wilting | cover and cook for 10-15 minutes over medium-low heat | you can cook it for longer if the pasta still isn't ready , over a lower flame | meanwhile , when the water in the pot starts boiling , add the pasta and cook until firm to the bite | when it is almost done , add the crumbled feta to the sauce , keeping the heat on low | when the pasta is ready , scoop it up with a slotted spoon or strainer , holding it over the pot to drain , and then add the drained noodles in batches to the sauce in the skillet , mixing thoroughly | cook the finished pasta on low heat for a few minutes | then add a grating of parmesan , if you like , and a few twists of freshly ground black p  
difficulty: 4.15 (easiest, 5 hardest)  
Please rate this recipe (1-5), or press Enter to skip: 5  
Any comments? (optional): perfect

Recipe 2:  
name: mushroom pasta  
cook\_time: 35 minutes  
ingredients: ['mushroom', 'butter', 'flour', 'milk', 'pasta', 'fontina cheese', 'salt and pepper', 'mugwepi']  
instructions: heat the oil in a large pot | add the mushrooms and cook until just browned | add the sliced mushrooms and saute until just browned | stir in the flour and cook for approx 3 minutes or until the b  
itter taste of the flour is gone | do not brown | add the hot milk and cook until just thickened | add the seasonings to your taste | combine the sauce with 2 cups of the cheese until melted | add to pasta in bowl and mix | spray a large casserole dish with non-stick cooking spray | cover and bake at 350F for 25 minutes or until bubbling  
nutrition: {'calories': 722.1, 'protein': 42.6, 'fat': 53.4, 'carbohydrates': 27.6}  
difficulty: 4.83333333333334 (massacc, 5 hardest)  
Please rate this recipe (1-5), or press Enter to skip: 4  
Any comments? (optional): high caloric

Recipe 3:  
name: bucatini alla liguria  
cook\_time: 40 minutes  
ingredients: ['olive oil', 'onion', 'garlic clove', 'tomatoes with juice', 'salt and pepper', 'garlic cloves', 'hot red pepper flakes', 'extra virgin olive oil', 'mixed nuts', 'fresh mint leaves', 'pepper', 'salt', 'pasta', 'mint leaf', 'pecorino cheese']  
instructions: for tomato sauce: heat oil and saute onion until translucent | add garlic and tomatoes with their juices , breaking up the tomatoes with a wooden spoon | season with salt and pepper to taste | simmer uncovered until most of the juices have evaporated  
for pesto: heat oil in a food processor or blender | combine garlic , hot pepper flakes , olive oil , nuts and mint | pulse to make a coarse paste | season with black pepper to taste and set aside | for pasta: bring a large pot of lightly salted water to a boil | add bucatini and cook until al dente | set aside 1 / 2 cup pasta water and drain pasta | return pasta to the pot and add pesto and tomato sauce | toss well to coat , adding reserved water as needed to thin sauce | garnish with fresh mint and serve with pecorino cheese  
nutrition: {'calories': 844.0, 'protein': 51.0, 'fat': 55.0, 'carbohydrates': 36.0}  
difficulty: 4.7000000000000005 (easiest, 5 hardest)  
Please rate this recipe (1-5), or press Enter to skip: 4  
Any comments? (optional): high calorie

Recipe 4:  
name: pasta with spinach feta and olives  
cook\_time: 25 minutes  
ingredients: ['olive oil', 'garlic', 'vinegar', 'garlic cloves', 'pasta', 'spinach', 'kalamata olive', 'capers', 'feta cheese', 'pepper']  
instructions: heat oil in a large bowl | add olive oil , vinegar and garlic | add remaining ingredients and toss to mix | the spinach will wilt from the hot pasta | season with pepper to taste | garnish with more cheese crumbles if desired | though it's delicious refrigerated  
nutrition: {'calories': 844.0, 'protein': 53.0, 'fat': 33.0, 'carbohydrates': 40.0}  
difficulty: 4.7000000000000005 (easiest, 5 hardest)  
Please rate this recipe (1-5), or press Enter to skip: 4  
Any comments? (optional): healthy

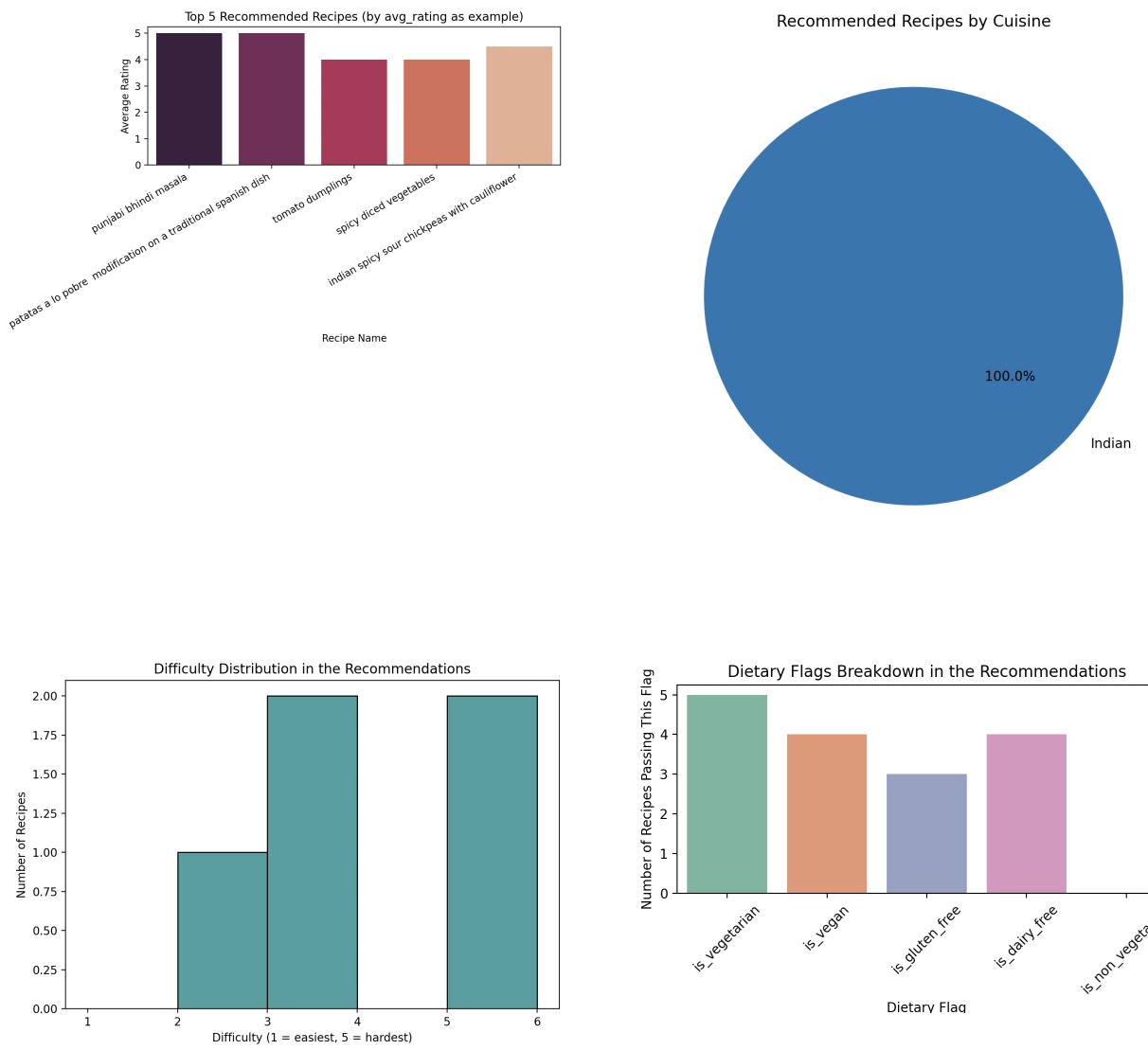
Recipe 5:  
name: simple stove top macaroni and cheese  
cook\_time: 40 minutes  
ingredients: ['olive oil', 'white onion', 'butter', 'flour', 'milk', 'stone ground dijon mustard', 'cayenne pepper', 'sharp cheddar cheese', 'salt', 'fresh parsley']  
instructions: In a large saute pan , over medium heat , cook the onions in the butter until softened | add the flour and cook for 3 minutes | add the mustard and cayenne | continue cooking for another 2 minutes | pour in the hot milk , whisk and simmer until thickened | stir in the cheese | taste for seasonings | combine the sauce and the macaroni , along with the parsley | let stand 5 minutes off the heat with the lid on | serve  
nutrition: {'calories': 910.0, 'protein': 50.0, 'fat': 38.0, 'carbohydrates': 66.0}  
difficulty: 3.6666666666666665 (hardest, 5 hardest)  
Please rate this recipe (1-5), or press Enter to skip: 3  
Any comments? (optional): high calorie  
User 5849 seems concerned about calories (from reviews).  
User 5849 seems concerned about healthy or nutrient aspects.  
Updated user prefs after NLP extraction: {user\_id: 5849, 'cuisine\_preference': 'Mexican', 'dietary\_restrictions': ['vegetarian'], 'available\_time': 50, 'available\_ingredients': ['pasta'], 'max\_difficulty': 5.0, 'nutrition\_preferences': {'calories': 400, 'protein': 20, 'fat': 10.0}}

## 2. Preference

### 3. Please enter your user preferences.

Enter a user ID (integer): 6849  
Enter cuisine preference (e.g., 'Indian', 'Italian', 'Mexican', 'Chinese', 'American'): Indian  
Enter dietary restrictions (comma-separated, e.g. 'vegetarian, non\_vegetarian, gluten\_free', 'vegan'): vegetarian  
Enter available time in minutes (integer): 45  
Enter available ingredients (comma-separated, e.g. 'tomatoes, onions, etc..'): onions, tomatoes  
Enter max difficulty (1-5, float allowed): 5  
Enter target calories (float): 1000  
Enter target protein in grams (float): 30  
Enter target fat in grams (float): 20

## Visualizations



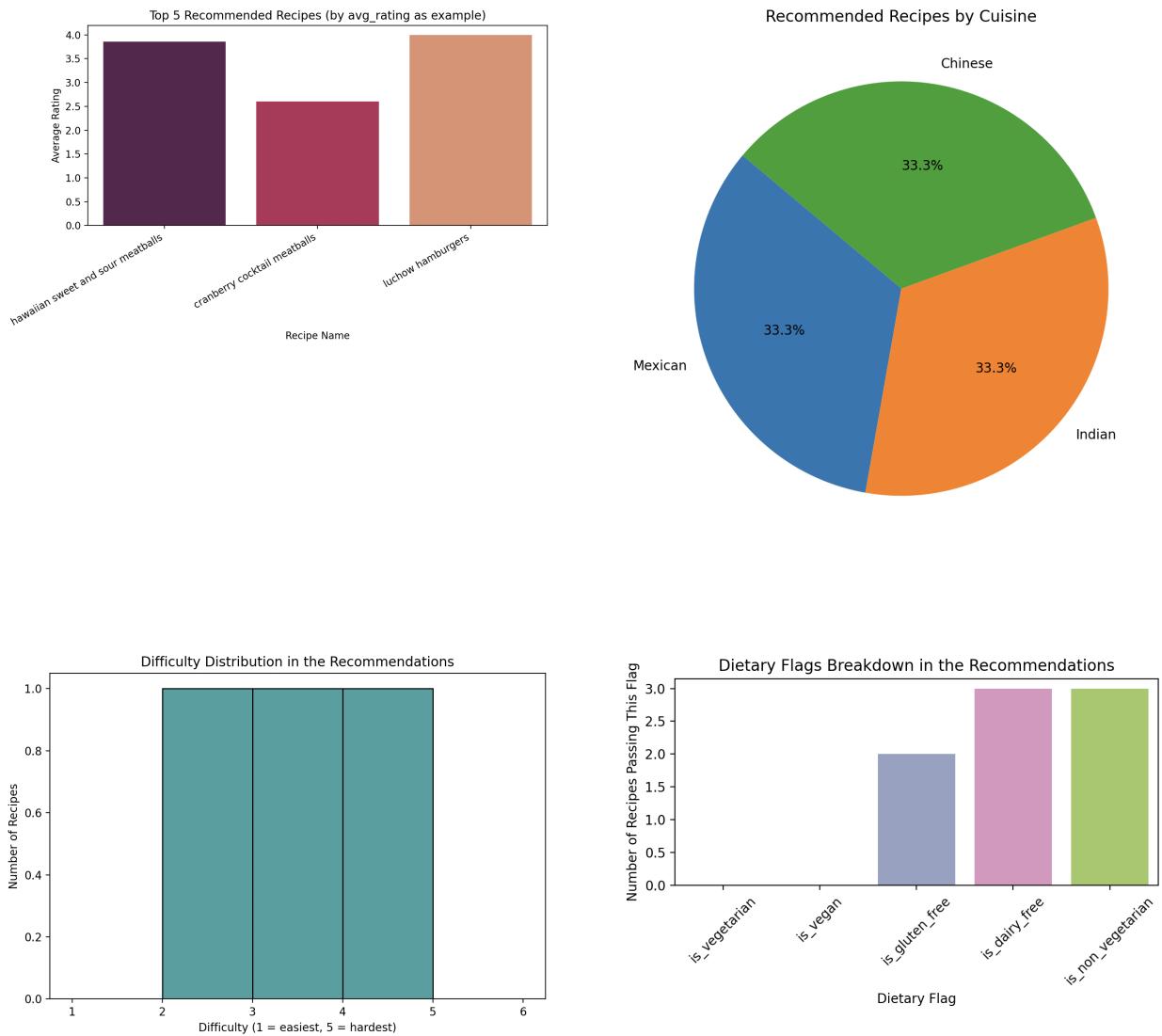
**The top 5 recommendations and modified user preference using NLP based on the comments given to the recipe by the user**

5. Top Recommendations:	
Recipe 1: name: punjabi bhindi masala cook_time: 10 minutes	instructions: ['fry onions', 'onions', 'tomatoes', 'ginger paste', 'coriander powder', 'turmeric powder', 'red chilli powder', 'jeera seeds', 'coriander leaves', 'oil', 'salt']   remove the bhindi with a moist cloth   cut bhindi in about 1" pieces   make a salt in each of the pieces so the masala gets applied   heat oil in kada   add jeera seeds   then add the cut onion and fry for some time till they become golden brown   the tomato pieces and let them cook   add ginger paste and coriander powder and fry for a minute   add turmeric powder, red chilli powder, mix it   this should become homogenous mixture which leaves out the oil   add the okra pieces to the mixture and appl the masala powder   let it cook on medium flame while stirring in between so that it does not stick to the kada   this may take 10 minutes   add salt and garnish with coriander leaves
difficulty: 3.6666666666666667 (1easiest, 5hardest)	
Please rate this recipe (1-5), or press Enter to skip: 5	
Any comments? (optional): perfect	
Recipe 2: name: patatas la lo pobre modification on a traditional spanish dish cook_time: 45 minutes	instructions: ['green peppers', 'red peppers', 'tomatoes', 'Water', 'vinegar', 'brown sugar', 'soy sauce', 'potatoes', 'onions', 'garlic', 'red chili pepper', 'cayenne pepper', 'tabasco sauce', 'pepper', 'salt', 'olive oil', 'oil', 'bread']   instructins: cut potatoes into smallish chunks and boil them for 4 or 5 minutes in salted water   they should be very firm and only slightly cooked   drain the potatoes and rinse with cold water   heat frying oil to 180c and fry for 5-8 minutes , or until they are cooked   not crispy   turn them up to medium heat and fry until golden and crisp   add the vegetables   add the green and red peppers   add the onions   add the garlic   add the red chili pepper very very carefully into really tiny pieces   add to the pan and wash your hands ! ! i mean ! ! go wash 'em and then come back !   i think in your peppers and your spices , you'll have to experiment with this , because i love my papas to be really spicy , i add about 2 tbs tabasco   a good tbsp peper and then a liberal amount of cayenne to give that extra colour and flavour   mix the water , soy sauce , vinegar and sugar and stir into the pan   i think this is the best part of this dish   in this jam , if you have just made it looks , umm , not too pleasant , but it tastes great   if you have left overs , refrigerate and then when you want to use it shape them into a sort of "party" by mashing the potatoes a bit , then either fry in oil or "dry-fry" them in a large pan
difficulty: {calories: 785.6, protein: 34.6, fat: 44.6, carbohydrates: 40.6}	
difficulty: {calories: 785.6, protein: 34.6, fat: 44.6, carbohydrates: 40.6}	
Please rate this recipe (1-5), or press Enter to skip: 3	
Any comments? (optional):	
Recipe 3: name: tomato dumplings cook_time: 40 minutes	instructions: ['onions', 'salt', 'flour', 'Milk', 'Flour', 'Oil']   instructions: combine first 5 ingredients together   let ball for about 5 minutes   then turn down heat , cover , and let simmer while making dumplings   combine the flour and milk to make dumplings   it will be sticky   drop into tomato pot , by teaspoon   cover a lid and stand , on low heat , for about 30 minutes or more   do not open lid until time is up
difficulty: {calories: 226.0, protein: 26.4, fat: 8.6, carbohydrates: 19.0}	
difficulty: {calories: 226.0, protein: 26.4, fat: 8.6, carbohydrates: 19.0}	
Please rate this recipe (1-5), or press Enter to skip: 4	
Any comments? (optional): low calorie	
Recipe 4: name: Spicy diced vegetables cook_time: 45 minutes	instructions: ['onions', 'potatoes', 'tomatoes', 'green beans', 'vegetable oil', 'black mustard seeds', 'onions', 'garlic cloves', 'fresh ginger', 'ground cardam', 'turmeric powder', 'dill powder', 'garam masala', 'salt']   instructions: peel carrots , potatoes and tomatoes and dice   trim beans and cut into small pieces   heat oil in a heavy based saucenan and fry mustard seed until they pop , add onion , garlic and ginger and fry stirring frequently until onion is golden   stir in tumeric , coriander , cumin and chilli powder and fry for a few seconds then toss in vegetables until coated with spices and oil   add salt to taste   add 3 tablespoons water , cover and cook for 15minutes or until vegetables are tender   stir gently every 5 minutes
difficulty: {calories: 223.0, protein: 10.0, fat: 12.0, carbohydrates: 21.0}	
difficulty: {calories: 223.0, protein: 10.0, fat: 12.0, carbohydrates: 21.0}	
Please rate this recipe (1-5), or press Enter to skip: 4	
Any comments? (optional): not healthy	
Recipe 5: name: Indian spicy sour chickpeas with cauliflower cook_time: 45 minutes	instructions: ['fresh ginger', 'lemon juice', 'vegetable oil', 'onions', 'jalapeno pepper', 'tomatoes', 'cumin', 'coriander', 'turmeric', 'cardamom', 'clove', 'salt', 'chickpeas', 'cauliflower floret', 'cilantro', 'water']   instructions: combine lemon juice and grated ginger in a small cup and set aside   heat a large saucpan over medium-high heat and add oil   saute onion and cook until onion becomes brown   add jalapeno and continue cooking until onion is brown all over   add torn cilantro and cardamom   add turmeric and cumin   add chickpeas and cilantro   add water   cover and cook over medium heat for approximately 10 minutes , stirring occasionally and adding additional water if mixture becomes dry   in a separate pan , combine cauliflower and 1/4 cup water   cover and cook over medium heat until cauliflower is tender   uncover cauliflower and boil off any excess liquid   remove from heat and stir in lemon juice mixture   combine cauliflower with chickpea mixture and add in chopped cilantro   serve with a green salad and naan bread or as a side dish
difficulty: 3.95 (1easiest, 5hardest)	
Please rate this recipe (1-5), or press Enter to skip: 5	
Any comments? (optional): perfect	
User 6849 sees concerned about calories (from review).	
Updated user prefs after NLP extractions: {user_id: 6849, 'cuisine_preference': 'Indian', 'dietary_restrictions': ['vegetarian'], 'available_time': 45, 'available_ingredients': ['onions', 'tomatoes'], 'max_difficulty': 5.0, 'nutrition_preferences': {calories: 40.0, protein: 30.0, fat: 40.0}}	

### 3. Preference

```
3. Please enter your user preferences.  
Enter a user ID (integer): 9527  
Enter cuisine preference (e.g., 'Indian', 'Italian', 'Mexican', 'Chinese', American'): American  
Enter dietary restrictions (comma-separated, e.g. 'vegetarian, non_vegetarian, gluten_free', 'vegan'): non_vegetarian  
Enter available time in minutes (integer): 50  
Enter available ingredients (comma-separated, e.g. 'tomatoes, onions, etc..'): beef, eggs  
Enter max difficulty (1-5, float allowed): 5  
Enter target calories (float): 1500  
Enter target protein in grams (float): 24  
Enter target fat in grams (float): 14
```

# visualizations



**The user selected American but the model recommended dishes from Indian, Mexican and Chinese cuisine because there were no American dishes as per the user's constraints. Hence the model recommended three closest dishes using the fallback mechanism.**

5. Top Recommendations:

Recipe 1:  
name: asian sweet and sour meatballs  
cook\_time: 30 minutes  
instructions: ['beef', 'cornstarch', 'nutmeg', 'salt', 'eggs', 'onion', 'pepper', 'garlic clove', 'brown sugar', 'fresh pineapple chunks', 'green pepper', 'garlic powder', 'pineapple juice', 'soy sauce', 'water', 'vinegar']  
method: ['add beef, cornstarch, onion, salt, eggs, garlic powder, brown sugar to meatballs | mix well and form into meatballs | heat oil in skillet | brown meatballs on all sides , or bake in oven until browned | in large saucpan add remaining cornstarch , soy sauce , vinegar , water , and brown sugar to pineapple juice | cook until thickened | stir constantly | add meatballs , fruit , and peppers | cook 5 minutes or until fruit is well heated | for档次 make meatballs as directed and flash freeze | transfer to a ziploc bag , label and put back into freezer | to serve prepare sauce as directed and add meatballs and heat through']  
nutrition: {'calories': 369.0, 'protein': 38.0, 'fat': 189.0, 'carbohydrates': 26.0}  
difficulty: 4.5 (1=easiest, 5=harsdest)

Please rate this recipe (1-5), or press Enter to skip:  
Any comments? (optional):

Recipe 2:  
name: cranberry cocktail meatballs  
cook\_time: 0 minutes  
instructions: ['corn flakes', 'eggs', 'parsley', 'ketchup', 'onions', 'soy sauce', 'garlic powder', 'pepper', 'cranberry sauce', 'chili sauce', 'brown sugar', 'lemon juice']  
method: ['in a large bowl , combine ground beef , cornflakes crumbs , parsley , eggs , ketchup , onion , soy sauce , garlic powder and pepper | mix well and form into small balls , from 1 / 2" to 3 / 4" in diameter | place in a casserole or baking pan | heat oven to 380 degrees f | meanwhile in a saucpan , combine cranberry sauce , chili sauce , brown sugar and lemon juice | cook stirring over medium heat until smooth | pour hot sauce over meatballs | bake for 30 to 45 minutes , depending on the size of the meatballs | once meatballs are done , keep warm and serve']  
nutrition: {'calories': 1264.1, 'protein': 35.0, 'fat': 189.0, 'carbohydrates': 17.0}  
difficulty: 3.1 (1=easiest, 5=harsdest)

Please rate this recipe (1-5), or press Enter to skip:  
Any comments? (optional):

Recipe 3:  
name: sandwich burgers  
cook\_time: 30 minutes  
instructions: ['beef', 'fat', 'white bread', 'salt', 'pepper', 'nutmeg', 'eggs']  
method: ['mix the meat and bread | squeeze as much water as you can out of the bread | add to the meat and mix smoothly | add the seasoning and eggs | combine well | shape into large patties | broil or cook in a little hot fat in a pan until brown and done | then add the meat to the bread | keep warm and serve']  
nutrition: {'calories': 1450.3, 'protein': 33.0, 'fat': 231.0, 'carbohydrates': 2.0}  
difficulty: 2.8 (1=easiest, 5=harsdest)

Please rate this recipe (1-5), or press Enter to skip:  
Any comments? (optional):  
[View Back/Reviews](#) found for user 9527.