Meal Match- A Personalised Recipe and Grocery Recommender for Sustainable Meal Planning

Nishant Jha (z5528738)
Department of Computer Science and Engineering, UNSW Sydney, NSW, 2052
Email: z5528738@ad.unsw.edu.au

1. Introduction

Meal Planning and grocery shopping are integral parts of daily lives, yet they pose several challenges for many individuals. The need to balance cooking skills, budget constrains and ingredient availability, often leads to food wastage and inefficiency. There are several recipe and grocery apps exist, but they fail to provide a personalised and practical recommendations. For example, current apps, such as Yummly and Allrecipe, rely on generic recommendations, prioritizing either sponsored products or highly rated recipes, and ignoring users' unique needs like specific nutritional goals or panty inventories. For instance, Yummly suggest recipes based on broad dietary filters (e.g., "vegetarian") but fails to account for ingredients already at home, leading to unnecessary purchases. Similarly, Instacart's grocery suggestions focus on user history or promotions, neglecting cooking skill levels or budget limits, which can overwhelm novice cooks. These shortcomings highlight the following major gaps in existing systems:

- Most systems fail to integrate user-specific factors like availability of ingredients, dietary requirements (e.g. low sodium products for hypertension) or irrelevant recommendations.
- Current apps rarely optimize food wastage reduction by prioritising recipes that use existing products from pantry.

Our system, **Meal Match** aims to address these challenges by delivering a highly personalized recipe recommendations and optimized grocery lists, using **content-based filtering and nutritional analysis to ensure recommendations are** both practical and sustainable, making meal planning seamless and efficient. The following report clearly outlines the design, implementation and evaluation of the proposed methodology in the upcoming sections. Section 2 deals with the project scope, Section 3 deals with the Dataset Exploration, Section 4 explores the methodology and Section 5 explores the evaluation metrics to be used to evaluate the proposed recommender system.

2. Scope of the Project

The **domain** of the Meal Match recommender system is food and grocery shopping, focussing on a personalised recipe recommendation and an optimised grocery list generation. For example, An **user with a vegetarian diet and having ingredients like pasta, olive oil and tomatoes in his pantry, input these details and our recommender system recommends recipes like "Tomato Pasta" supplemented by a grocery list for missing items like Garlic, thereby ensuring minimal wastage of resources.**

2.1. Intended Users

The intended users of Meal Match are the individuals who face challenges in meal planning and grocery shopping. The following are the **Target User Types (Scope):**

- Busy professionals: Individuals with a limited time for meal planning and seeking a quick tailored meal plan and effective grocery lists. For example, An IT professional with a low-carb diet wants a quick dinner idea using ingredients such as eggs and spinach.
- Families: households cooking for multiple individuals and needing a budget friendly recipe. For example, a parent cooking for a family of four with a 60AUD weekly budget, seeking recipes balancing both nutritional values and cost.
- Novice Cooks: A college student, new to cooking needs, looking for a basic beginner friendly recipe using ingredients like rice, beans.
- Users with specific dietary needs like vegan, gluten-free, low-sodium diet for a patient suffering from diabetes and hypertension.

Meal Match shows 3 to 5 recipes at a time on the recipe results screen to avoid overwhelming users while providing sufficient variety. This number is based on the practices used in recommender systems, balancing choice with decision fatigue. For each recipe, details like, cooking time, ingredients, difficulty level are displayed. If a user selects the recipe, a consolidated grocery list is displayed listing all missing ingredients.
For example, after inputting vegetarian, beginner, pantry: rice, carrots, the user sees:

- Recipe 1: Carrot Fried Rice (uses rice, carrots; needs soy sauce)
- Recipe 2: Veggie Soup (uses carrots; needs broth)
- Recipe 3: Rice Salad (uses rice; needs cucumber, vinegar)
- The grocery list might display the following: soy sauce, vegetable broth, cucumber, vinegar.

2.2. User Interface Design

Meal Match is designed as a **web-based application**. The interface is built using HTML, CSS and JavaScript for the frontend and Flask for the backend. Key UI features are as follows:

- Input Form: Users enter dietary preferences, skill level, pantry items, and budget via checkboxes, radio buttons, and text fields.
- Recipe Results: Displayed as scrollable cards, each showing recipe details and a 5-star rating system for feedback.
- Grocery List: A clean list of missing ingredients with action buttons like save, email.
- Star ratings and comments collected post-recipe viewing or use for feedback.

<u>An example interaction is given as</u>: a user opens the Meal Match website and enter the following details, vegan, intermediate, pantry: tofu, broccoli," and sees 4 recipe cards. They rate a "Tofu Stir-Fry" recipe 4 stars and generate a grocery list for soy sauce and garlic.

2.3. Comparison with other platforms

Table 1. Comparative analysis of our model with other applications.

Feature	Meal Match (Ours)	Yummly	Allrecipes	Instacart
Personalized Recipe Recommendations	Yes, based on dietary preferences, skill level, pantry, and budget.	Yes, based on diet and preferences, but limited pantry integration.	Yes, based on diet, but no pantry or skill-level filtering.	No, focuses on grocery delivery, not recipes.
Pantry-Based Suggestions	Prioritizes recipes using user's existing ingredients to reduce waste.	Limited; requires manual input of exclusions.	No pantry-based filtering.	No pantry integration.
Nutritional Filtering	Uses USDA dataset for precise dietary alignment (e.g., low-carb, vegan).	Basic nutritional filters (e.g., calorie range).	Limited nutritional data; relies on user input.	No nutritional filtering for recipes.
Grocery List Generation	Generates optimized list of missing ingredients with estimated costs.	Basic list generation, not optimized for pantry.	Manual list creation from recipes.	Auto-generates lists based on cart, not recipes.
Food Waste Reduction	Explicit focus via pantry prioritization.	No explicit focus.	No focus on waste reduction.	No focus on waste reduction.

2.4. Mock Ups and User Interaction

The Input Form Screen as shown in figure 1 allows users to specify dietary preferences (e.g., vegetarian, low-carb), cooking skill level (beginner, intermediate, advanced), pantry inventory (e.g., "tomatoes, pasta"), and an optional weekly budget via a responsive web interface. Users interact by checking boxes, selecting radio buttons, entering text, and clicking a "Find Recipes" button to generate personalized recipe recommendations. Feedback is indirectly collected by allowing users to edit their inputs post-recommendation (via an "Edit Preferences" button on the results screen), which updates the system's understanding of preferences and refines future suggestions through iterative learning.

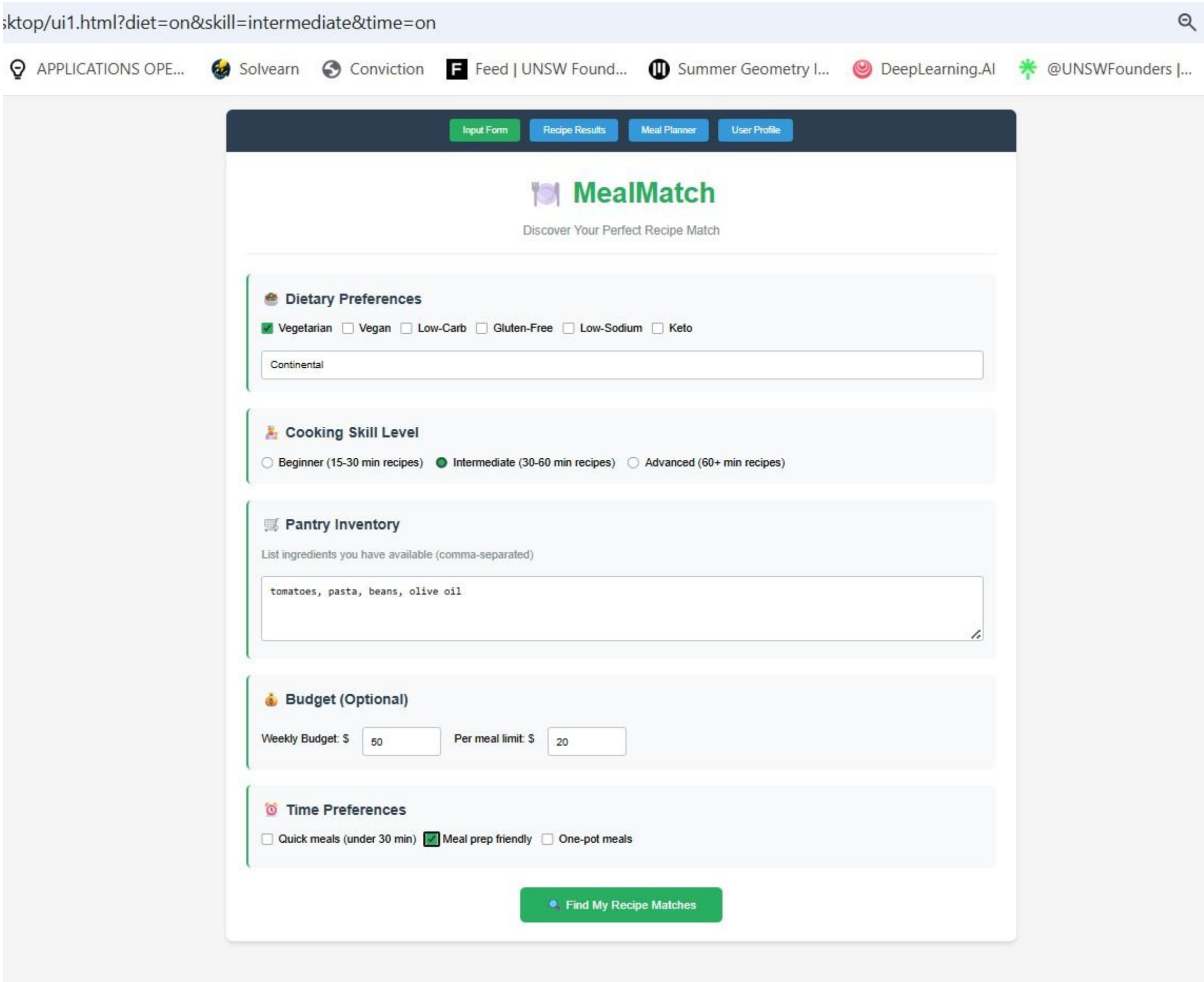
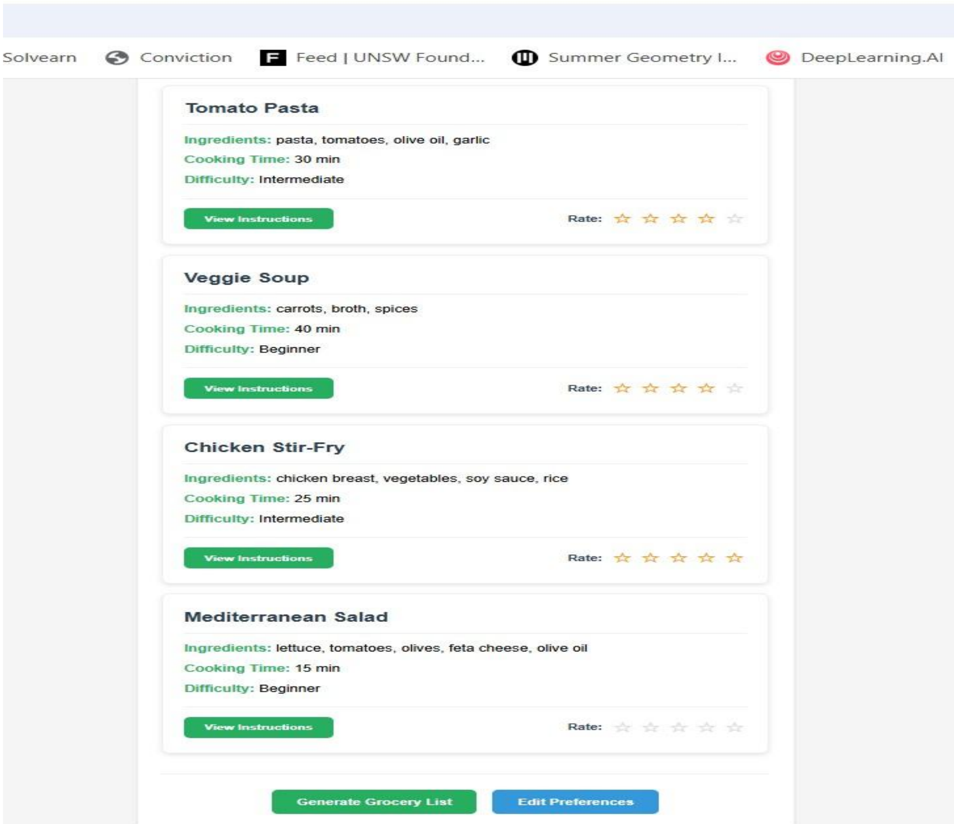


figure 1. Mockup of input from screen.



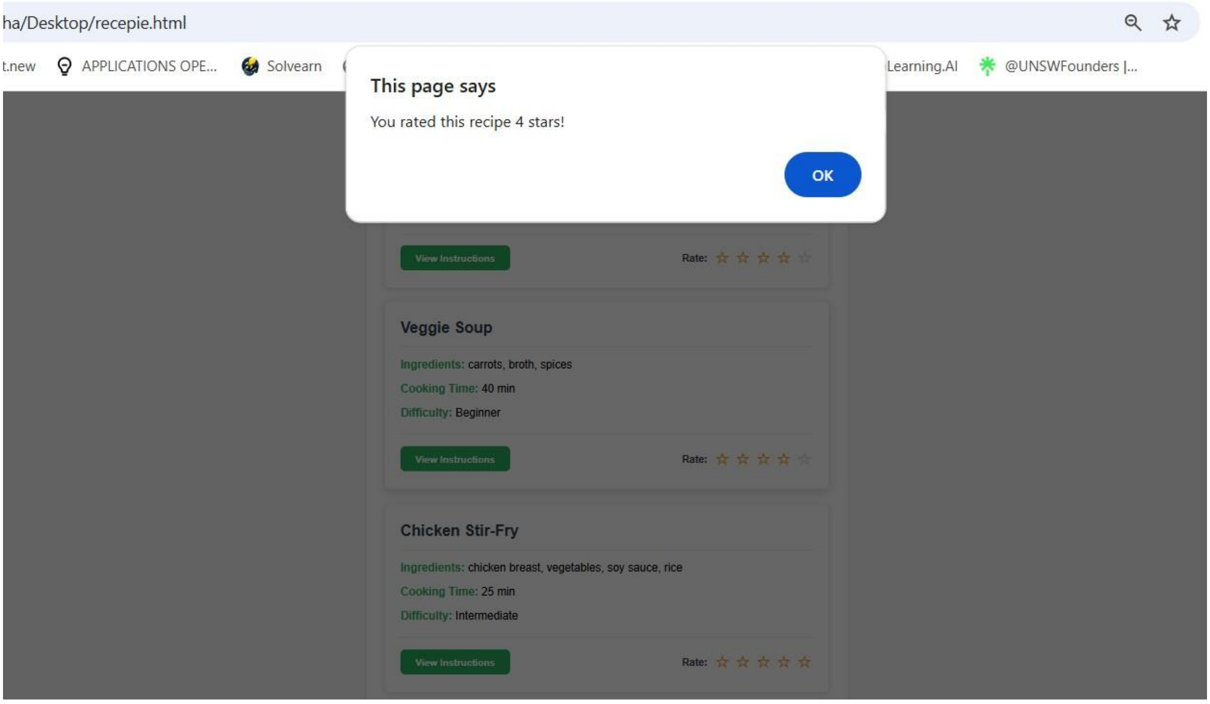


Figure 2. Mockup of Recipe Result Screen with Ratings.

The Recipe Results Screen, given by figure 2 displays 3-5 personalized recipe recommendations, presented as scrollable cards with details like recipe title (e.g., "Tomato Pasta"), ingredients, cooking time, and difficulty. Users interact by viewing recipe instructions via a "View Instructions" button and providing feedback through a 5-star rating system directly on each card. Feedback is collected via these ratings, which are stored in an SQLite database and used to adjust recipe weights in the content-based filtering algorithm, enhancing future recommendations based on user preferences.

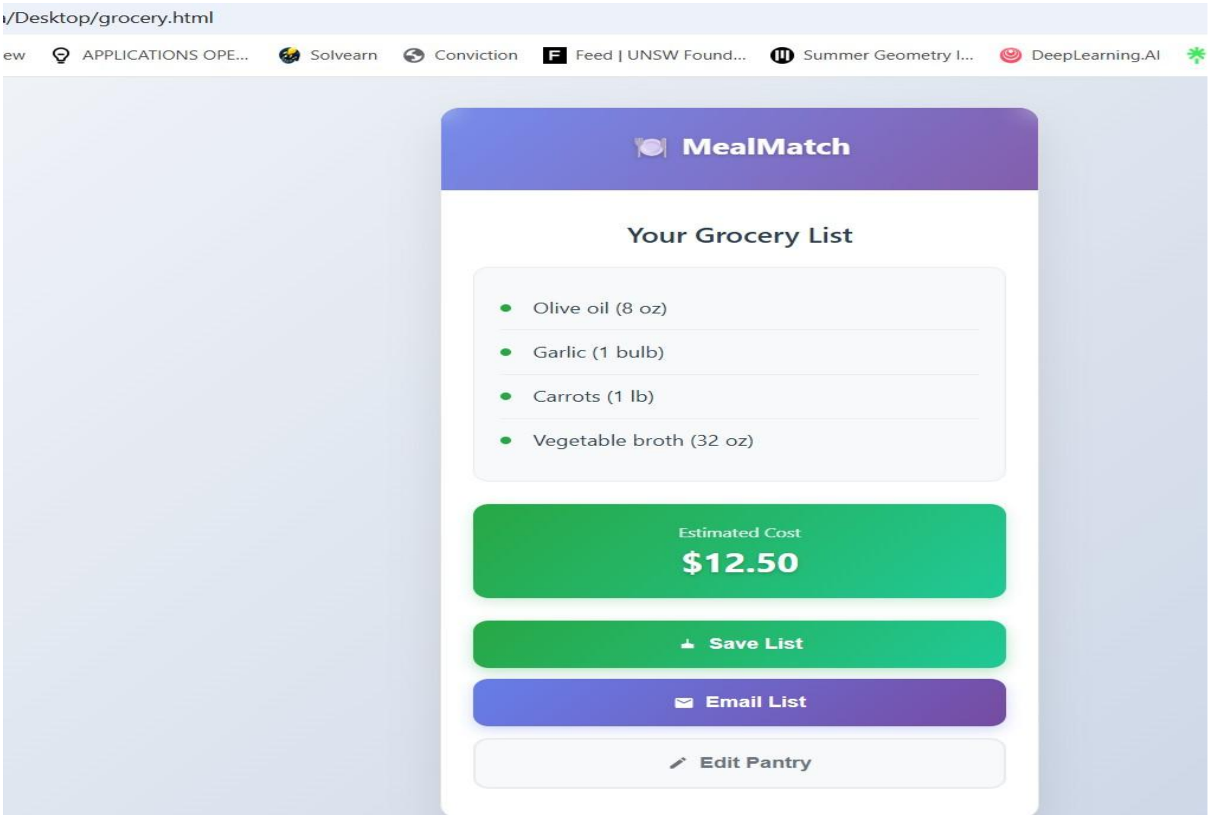


Figure 3. Mockup of grocery list screen

The Grocery List Screen given by figure 3, shows a concise list of missing ingredients (e.g., "olive oil, garlic") for selected recipes, along with an estimated cost, Users interact by saving the list as a text file, emailing it, or editing their pantry via dedicated buttons. Feedback is obtained indirectly when users update their pantry inventory post-shopping, which refines the system's ingredient-matching accuracy and improves subsequent grocery list recommendations by learning from updated user data.

2.5. User Study and Cold Start Problem

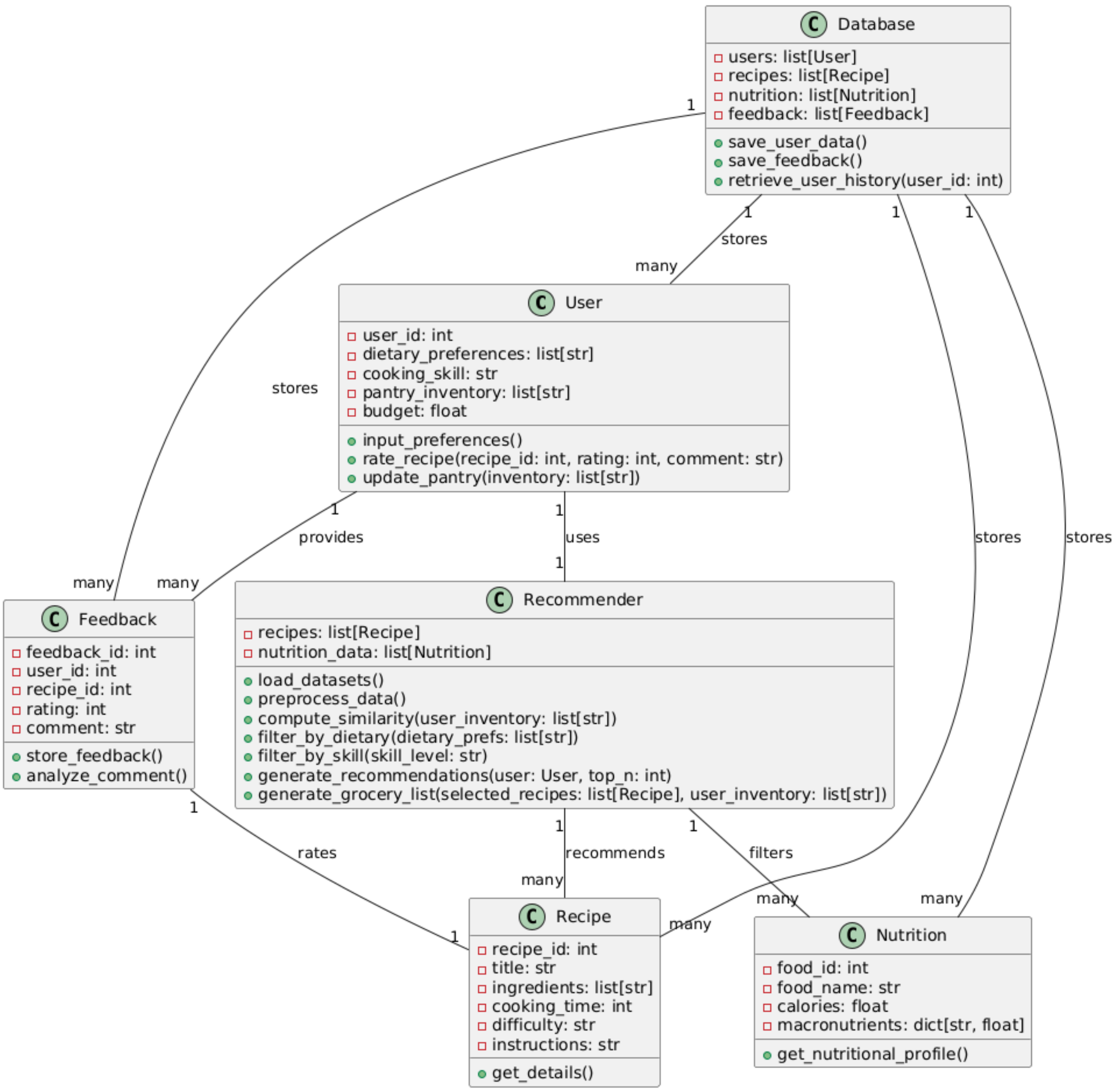


Figure 4. UML Diagram to show dynamic user interaction and study. This figure is drawn using PlantUML Online(https://editor.plantuml.com/)

- The UML diagram given in figure 4 for the MealMatch recommender system illustrates user interaction through the User class, which stores preferences (dietary, cooking skill, pantry, budget) and interacts with the Recommender class to input data and receive personalized recipe recommendations. Users provide feedback via the Feedback class (ratings and comments), which is stored in the Database class for future personalization. The Recommender class processes inputs using the Recipe (from RecipeNLG) and Nutrition (from USDA) classes to generate recommendations and grocery lists. The Database class ensures persistence of user data and feedback, enabling seamless interaction across sessions.

- The Meal Match recommendation model updates dynamically to enhance personalization. User feedback, collected as 5-star ratings and optional comments on the Recipe Results Screen, is stored in an SQLite database via the Feedback class. Weekly or after 100 feedback entries, the model retrains by adjusting recipe weights in the content-based filtering algorithm (using TF-IDF for ingredient similarity), prioritizing high-rated recipes. Comments are analysed with NLTK to extract preferences (e.g., avoiding spicy recipes), and pantry updates refine ingredient matching, ensuring recommendations evolve with user needs.

- Meal Match addresses cold start problems for new users and recipes effectively. For user cold start, a default profile based on the National Household Food Consumption Survey (available on Kaggle) is used, prompting new users to input basic preferences (e.g., vegetarian, pantry items) to generate initial recommendations. For item cold start, new recipes in the RecipeNLG dataset are preprocessed with TF-IDF vectors and tagged with USDA nutritional data, making them instantly recommendable. A hybrid approach combines content-based filtering with popularity-based recommendations (from common RecipeNLG recipes) until sufficient user feedback personalizes the system further.

2.6. Business Model and Revenue Generation

Meal Match's business model combines a freemium approach with strategic partnerships to generate revenue while maintaining accessibility. The core web-based recommender system is free, offering personalized recipe and grocery list recommendations to attract users. Revenue is generated through premium subscriptions (A$5/week) that unlock advanced features like meal prep scheduling, nutritional analysis, and integration with grocery delivery services. Additionally, affiliate partnerships with online grocery retailers (e.g., Amazon Fresh) earn commissions when users purchase recommended ingredients via linked stores. Sponsored recipe placements from food brands (e.g., promoting a specific pasta brand in recipes) provide further revenue, ensuring recommendations remain unbiased and user-focused while monetizing the platform's high engagement.

3. Dataset

To build a robust Personalized Grocery Shopping and Recipe Recommender System for the Meal Match , we need datasets of sufficient quality and quantity to support personalized recipe recommendations and grocery list generation based on dietary preferences, cooking skills, pantry inventory, and budget constraints. For our model we are using the following datasets,
RecipeNLG Dataset available at (https://www.kaggle.com/datasets/saldenisov/recipenlg) and USDA Food Composition Database available at (https://www.kaggle.com/datasets/econdata/usda-food-composition-databases)

3.1. About RecipeNLG Dataset Quality and Quantity and How it will be used:

The RecipeNLG Dataset, developed by the Poznań University of Technology, is a comprehensive collection of over **2.2 million recipes** scraped from various online sources. It is publicly available upon request from RecipeNLG and provided in CSV or JSON format. The dataset includes fields such as title (recipe name), ingredients (list of required ingredients), directions (cooking instructions), and NER (named entity recognition for ingredients). This dataset is ideal for generating personalized recipe recommendations based on user pantry and preferences. With over 2.2 million recipes, the dataset offers a vast pool for recommendation, covering a wide range of cuisines, dietary types, and skill levels, enabling robust personalization even for niche preferences (e.g., vegan keto recipes). This is useful for our recommendation as follows:

- title: Used to display recipe names to users (e.g., "Tomato Pasta").
- ingredients: Matched against user pantry inventory using TF-IDF vectorization to compute similarity scores, prioritizing recipes that maximize ingredient reuse.
- directions: Provided to users when they click "View Instructions" for detailed cooking steps.
- NER: Ensures accurate ingredient matching by standardizing terms, critical for aligning user inputs with recipe requirements.

For example, A user inputs "vegetarian, beginner, pantry: rice, carrots." The system uses TF-IDF on the ingredients field to find recipes like "Carrot Fried Rice" (ingredients: rice, carrots, soy sauce) with a high similarity score. The title ("Carrot Fried Rice") and directions ("Cook rice, stir-fry with carrots…") are displayed, and the NER field ensures "carrots" matches across variations (e.g., "baby carrots").

3.2. About USDA Food Composition Database Quality and Quantity and How it will be used:

The USDA Food Composition Databases, specifically the Food Data Central Foundation Foods dataset, is a high-quality nutritional dataset provided by the U.S. Department of Agriculture. It is publicly available for download from USDA FoodData Central in CSV format. The dataset includes fields like fdc_id (unique food identifier), description (food name), food_nutrients (nutrient details like calories, protein, carbs), and food_category (e.g., vegetables, grains). This dataset is critical for filtering recipes by dietary needs and ensuring nutritional alignment. This dataset Covers over 7,000 food items in the Foundation Foods subset, sufficient to map to most ingredients in the RecipeNLG dataset, supporting a wide range of dietary requirements (e.g., low-carb, gluten-free). This will be used in our recommendation system as follows:

- description: Maps food names to RecipeNLG ingredients (e.g., "tomatoes" to "tomato, raw").
- food_nutrients: Filters ingredients by nutritional constraints (e.g., <10g carbs per serving for low-carb diets).
- food_category: Groups ingredients for dietary alignment (e.g., exclude "meat" for vegetarians).
- fdc_id: Ensures unique identification for mapping and tracking.

For example, A user selects low-carb and inputs chicken, spinach. The system maps chicken to chicken, breast, raw (fdc_id: 171477) and spinach to spinach, raw (fdc_id: 168462) in the USDA dataset, checking that both have low carbs (<5g per serving). It then filters RecipeNLG recipes like Chicken Spinach Salad that align with these nutritional constraints.

3.3. Limitations of These Datasets

1. The RecipeNLG Dataset, while containing 2.2 million recipes, represent only a subset of recipes from original websites, potentially missing niche or region-specific dishes, which limits content-based filtering's diversity and can cause TF-IDF methods to underperform for users seeking unique cuisines (e.g., underrepresented African recipes).
2. Both datasets are pre-processed (e.g., RecipeNLG's NER standardization, USDA's lab-based nutritional data), omitting real-world complexities like user-specific ingredient variations or cooking method impacts, making collaborative filtering less effective due to missing user interaction data and reducing recommendation realism.
3. RecipeNLG, is tailored for specific tasks (e.g., recipe generation), with predefined splits that encourage overfitting of models like TF-IDF or neural nets to static metrics (e.g., BLEU scores), which may not translate to dynamic, user-centric recommendation performance in Meal Match.
4. Neither dataset includes user behavior or feedback data (e.g., ratings, preferences), forcing reliance on content-based filtering over collaborative filtering, which struggles to personalize without historical interactions and exacerbates cold start problems for new users.
5. Both datasets suffer from inconsistent ingredient naming (e.g., "tomato" vs. "tomatoes, raw" in RecipeNLG and USDA), necessitating fuzzy matching, which can introduce errors or computational overhead.
6. Neither dataset includes pricing information, limiting MealMatch's ability to accurately estimate grocery costs for budget-conscious users without external data supplementation.

4. Methods

4.1. Statistical Analysis of Data

Figure 5 represents the workflow of how the statistical analysis of data would be done to ensure high quality and relevant data should be used in our recommendation model, that is context aware. **Figure 5 is drawn using graphviz online tool, available at, https://dreampuf.github.io/GraphvizOnline**

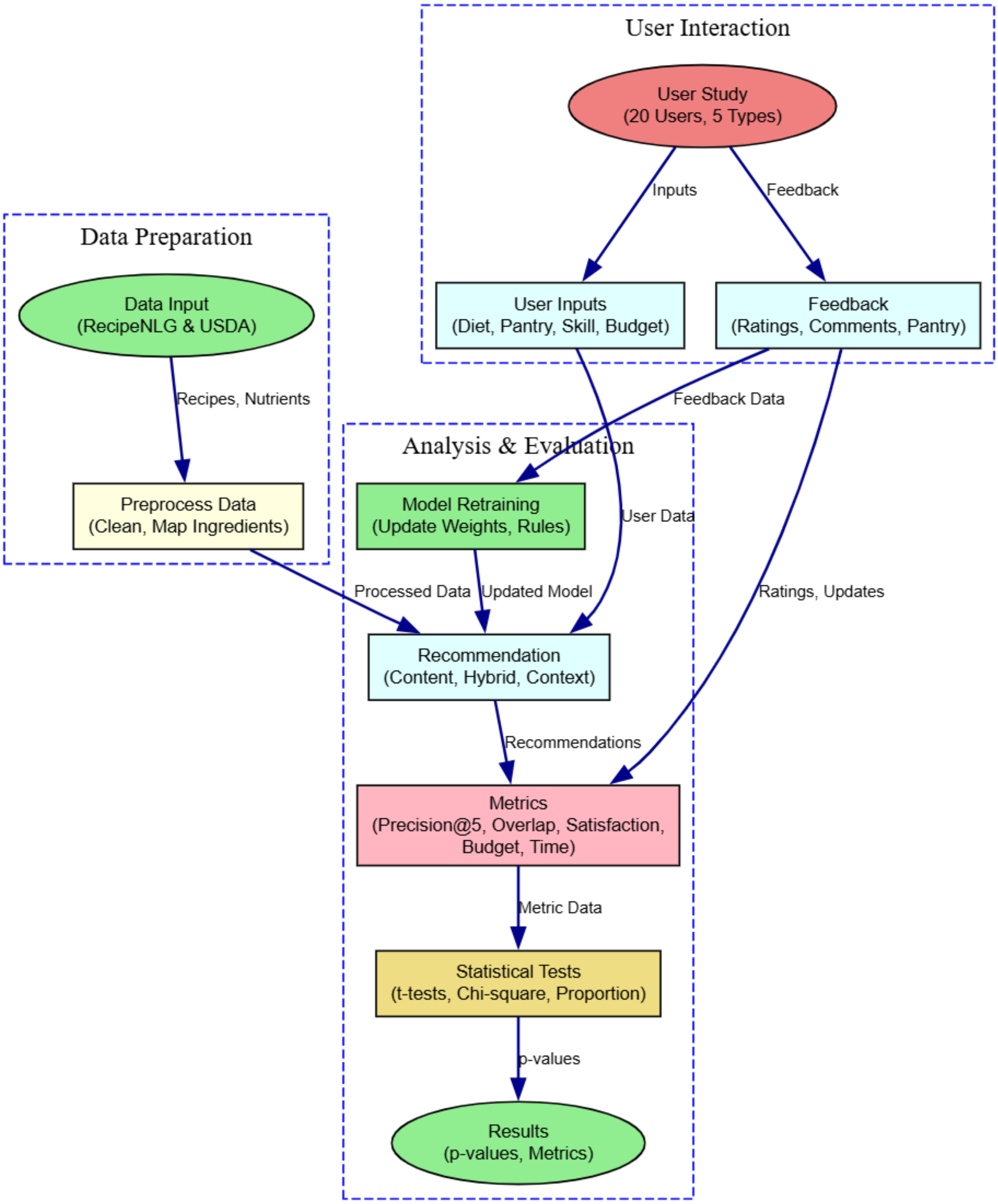


Figure 5. Illustration of Workflow of Method Proposed

The proposed system employs statistical analysis techniques to evaluate its performance in delivering personalized recipe and grocery list recommendations using the RecipeNLG and USDA Food Composition Datasets. Key metrics include Precision@5 (recipe relevance), Ingredient Overlap (pantry utilization), User Satisfaction (survey-based), Budget Adherence (hybrid method), and Time Relevance (context-aware method). Statistical tests, paired t-test for Precision@5 and Time Relevance, one-sample t-test for Ingredient Overlap, chi-square test for User Satisfaction, and one-sample proportion test for Budget Adherence, ensure robust evaluation.

4.2. Proposed Method

4.2.1. Content-Based Filtering (Primary Method)

We propose a content-based filtering method to deliver personalized recipe recommendations and optimized grocery lists. This method recommends recipes by matching user inputs (pantry inventory, dietary preferences, cooking skill level) to recipe attributes in the RecipeNLG dataset, augmented with nutritional filtering from the USDA dataset. It uses TF-IDF vectorization and cosine similarity to prioritize recipes with high ingredient overlap and dietary compliance.

<u>Key Points:</u>

- TF-IDF Vectorization: Converts ingredients (RecipeNLG) and user pantry into vectors to compute similarity scores, prioritizing recipes that reuse pantry items.
- Fuzzy Matching: Aligns RecipeNLG ingredients with USDA description fields (using fuzzywuzzy) to ensure accurate nutritional filtering.
- Nutritional Scoring: Filters recipes using USDA food_nutrients (e.g., calories, carbs) to match dietary preferences (e.g., low-carb <10g/serving).
- Difficulty Filtering: Uses RecipeNLG directions length or manual tagging to infer skill level (e.g., <10 steps for beginners).
- Grocery List Generation: Identifies missing ingredients and estimates costs

For example, a user inputting "vegetarian, beginner, pantry: rice, carrots" will have their pantry vectorized, matched to "Carrot Fried Rice" (similarity: 0.9, needs soy sauce), verified for vegetarian compliance using USDA food_category, and receive a grocery list for "soy sauce." This implementation is suitable because it leverages RecipeNLG's extensive recipe data and USDA's nutritional precision, addressing dataset limitations like missing user interaction data by relying on content-based techniques, and ensures efficient, scalable processing for diverse user needs (e.g. health-conscious dietary alignment) while supporting rapid evaluation through user studies.

4.2.2. Hybrid (Content-Based + Knowledge-Based) Filtering

To enhance personalization for our model, we will implement a hybrid recommendation method that integrates content-based filtering with knowledge-based rules to address strict user requirements, such as budget and dietary constraints, for our diverse user base, including families and health-conscious individuals. The content-based component will employ TF-IDF vectorization (using scikit-learn) to match user pantry inputs (e.g., "pasta, tomatoes") with RecipeNLG dataset `ingredients`, computing cosine similarity to prioritize recipes with high ingredient overlap, while the USDA Food Composition Databases will filter recipes using `food_nutrients` for dietary compliance (e.g., vegan excludes animal-based foods in `food_category`).

Knowledge-based rules will incorporate budget constraints by filtering recipes where missing ingredients' estimated costs (e.g., $1/oz for soy sauce, assumed or externally sourced) fit the user's budget, dietary rules using USDA `food_nutrients` (e.g., <10g carbs/serving for low-carb), and skill rules limiting recipes to step counts matching skill level (e.g., ≤10 steps for beginners, inferred from RecipeNLG `directions`).

A weighted scoring mechanism will combine TF-IDF similarity (70%) with rule-based compliance (30%), calculated as `score = 0.7 * TF-IDF_similarity + 0.3 * rule_compliance`, to rank the top 3-5 recipes, followed by grocery list optimization to minimize missing ingredients by prioritizing recipes with overlapping needs.

For example, a family inputting "vegetarian, intermediate, pantry: pasta, tomatoes, budget: $10" will receive recommendations for "Tomato Pasta" (TF-IDF: 0.95, needs garlic, $2) and "Veggie Pasta" (TF-IDF: 0.85, needs zucchini, $3), both verified as vegetarian via USDA data, with an optimized grocery list of "garlic, zucchini" totaling $5. This hybrid approach is suitable because it leverages RecipeNLG's extensive recipe pool for content-based matching and USDA's nutritional granularity for precise dietary filtration.

4.2.3. Context-aware content-based filtering method

We will implement a context-aware content-based filtering method, to enhance personalization by incorporating contextual factors like time of day and location, addressing the dynamic needs of busy professionals and eco-conscious users.
The method will use TF-IDF vectorization (scikit-learn) to match user pantry inputs (e.g., "chicken, spinach") with RecipeNLG dataset `ingredients` computing cosine similarity, and filter recipes using USDA Food Composition Databases `food_nutrients` for dietary compliance (e.g., low-carb <10g/serving).
Contextual features will include time-based filtering, prioritizing recipes with shorter cooking times (inferred from RecipeNLG `directions` length, e.g., <20 min for evenings), and location-based filtering, tagging USDA `description` fields for seasonal ingredients (e.g., tomatoes in summer) using external lookup tables.
Dynamic ranking will adjust TF-IDF scores (e.g., +0.2 for recipes <20 min at 6 PM), and NLTK-based NLP will analyze feedback comments (e.g., "too long") to refine time-based weights weekly. Grocery lists will prioritize locally available ingredients. For example, at 6 PM, a professional inputting "low-carb, beginner, pantry: chicken, spinach" will see "Chicken Spinach Salad" (15 min, TF-IDF: 0.9) prioritized over "Chicken Casserole" (45 min, TF-IDF: 0.85).

5. Evaluation Metrics

5.1. Metrics for Assessing the Recommendation Model

The following metrics evaluate the recommendation model and recommender system (user feedback/interactions), tailored to the presentation of 3-5 recipes per session and the diverse user base. We include **Top-N accuracy** (specifically Precision@5) for the model and system-level metrics like user engagement and click-through rates, leveraging dataset fields where possible.

- **Precision@5** measures the proportion of the top-5 recommended recipes rated ≥4 stars (1-5 scale) in a simulated dataset or historical feedback, reflecting recommendation relevance. It is suitable as a classic Top-N metric for the 3-5 recipe presentation, ensuring accuracy for users like health-conscious individuals needing dietary-compliant recipes. For example, a vegetarian user rating 4/5 recipes ≥4 stars (Precision@5 = 0.8) indicate high model accuracy. This metric is critical but requires simulated ratings due to RecipeNLG's lack of user interaction data.

- Ingredient Overlap calculates the ratio of recipe ingredients in the user's pantry, averaged across users, using RecipeNLG ingredients and NER fields. It evaluates pantry utilization, crucial for eco-conscious users, computed as (pantry ingredients in recipe) / (total recipe ingredients). For example, a recipe needing "rice, carrots, soy sauce" with pantry "rice, carrots" has an overlap of 0.67. This per-user metric is vital for sustainability but is limited by ingredient name inconsistencies, which is addressed via fuzzy matching.

- Click Through Rate (CTR) measures the proportion of recommended recipes users click to view detailed instructions (tracked via the "View Instructions" button on the Recipe Results Screen), reflecting engagement. For example, if users click 3/5 recipes, CTR = 0.6. This system-level metric uses interaction logs (stored in SQLite, not in datasets) and is critical for assessing user interest, especially for busy professionals seeking quick recipes, but requires interface tracking.
- Time Relevance Score (User Feedback): Time Relevance Score measures the percentage of recipes matching contextual time constraints, <20 min for evening meals, inferred from RecipeNLG *directions* length)

5.2. Computational and Other Requirements

- The system employs TF-IDF vectorization and cosine similarity (scikit-learn) for content-based filtering, with fuzzy matching (fuzzywuzzy) to align RecipeNLG ingredients with USDA description. These operations process 2.2 million recipes and 7,000+ foods in <1 second per query on a standard CPU (e.g., 2.5 GHz, 8GB RAM). For example, a user inputting "low-carb, pantry: chicken, spinach" triggers TF-IDF and filtering, delivering recommendations instantly via Flask. Precomputed TF-IDF matrices reduce runtime, ensuring near-real-time performance for the web interface.
- Weekly or after 100 feedback entries, the model retrains TF-IDF weights and knowledge-based rules using user ratings and NLP-analyzed comments (NLTK). Retraining takes ~10 minutes for 2.2 million recipes, updating weights for high-rated recipes (e.g., 4-star "Tofu Stir-Fry" boosts vegan recipes). SQLite stores feedback, enabling efficient updates without full reprocessing.
- Flask and SQLite scale to thousands of users, with low-latency recommendation generation. However, budget estimation relies on assumed costs (e.g., $1/oz for soy sauce), and location-based filtering (context-aware method) may need external APIs, adding latency unless precomputed. These address dataset limitations but require careful validation.

5.3. User Design and Feedback Solicitation

We will conduct an informal user study, not requiring ethics approval, to evaluate the recommender system with 30 real users (6 per user type: busy professionals, families, novice cooks, health-conscious, eco-conscious) through a simulated Flask-based web interface, collecting feedback via a questionnaire to assess usability, relevance, and engagement.

Study Design includes the following:

- **Participants**: 30 volunteers recruited via social media or university networks, ensuring diversity (e.g., vegan professionals, budget-conscious families).
- **Procedure**:
    1. Users access the web interface (Input Form, Recipe Results, Grocery List screens) on their devices.
    2. They perform 3 meal planning tasks: input preferences (e.g., "vegetarian, beginner, pantry: rice, carrots"), receive 3-5 recipes, rate them (1-5 stars), click "View Instructions" (tracked for CTR), and generate a grocery list.
    3. Tasks vary by user type (e.g., evening meals for professionals, budget-limited for families).
    4. Users complete a post-study questionnaire via a web form.
- **Duration**: Each session takes less than 20 minutes.

The feedback Questionnaire includes the following Questions:

Q1. On a scale of 1-5, how satisfied are you with the recommendations? (1 = very dissatisfied, 5 = very satisfied).

Q2. How well did the recipes match your dietary preferences and pantry? (1-5).

Q3. How easy was the interface to navigate? (1-5)

Q4. How likely are you to use the system again and Were the recipes suitable for your time constraints? (1-5)

Note: Quantitative data (e.g., mean Satisfaction = 4.2/5) is analyzed with statistical tests (e.g., chi-square for satisfaction across user types, $p < 0.05$ indicating differences).