Optimizing Nutrition: A Cost-Effective Meal Generator System

CS221 Project Progress Report — Nuria Perez Casas

November 15, 2024

Introduction

In this project, we tackle the challenge of generating cost-effective, nutritionally balanced weekly meal plans. The aim is to develop an AI system that produces menus within user-specified budgets while maintaining a balanced diet. The problem involves complex trade-offs, as lower-cost meals often lack nutritional diversity, and ingredient choices may limit recipe variety. To address these challenges, we focus on developing an algorithm that selects recipes optimized for cost, nutritional value, and avoidance of repetitive meals. This work is particularly relevant in the context of food deserts and lower-income households, where meal planning can be limited by access to affordable, healthy food options.

Literature Review

Wallingford and Masters (2023) [2]developed an educational tool using optimization and linear programming to help students analyze the health and cost impacts of food choices. They use Excel to visualize nutrient levels in low-cost diets at the ingredient level. While their work isn't a meal planner, it offers valuable insights on cost and nutrition balance that could complement my recipe-based approach. Comparing my algorithm's performance against their ingredient-focused model might reveal potential differences in cost-effectiveness or nutritional coverage when recipes are used as the basis.
A study on Cost Optimization for Weekly Meal Planning of College Students (2022) [1] aligns closely with my project goals, using a constraint satisfaction approach with calorie and nutrient limits tailored to students' gender and activity levels. Their research proves that balanced, affordable diets are achievable, but my focus shifts to making the process recipe-based and more practical for weekly planning. This makes our approaches complementary: while they confirm that balanced diets can be budget-friendly, I'm exploring how to achieve this with real recipes for ease of use.
Some online tools also offer weekly menu generation, but most don't let users set a budget and often come with a price tag. My project seeks to fill this gap, allowing for budget-friendly, nutritious meal planning that's both accessible and tailored to the user.
In summary, existing tools and studies provide useful frameworks for diet cost and nutrition optimization. My project builds on these by focusing on recipe-based planning that's easy to use and affordable, aiming to bring together cost, nutrition, and practical weekly menu generation in one place.

Dataset

Our dataset, sourced from Kaggle `https://www.kaggle.com/datasets/wilmerarltstrmberg/recipe-dataset-over-2m`, contains over 2 million recipes with information like name, source, ingredients, and directions. From this, we randomly selected 100 recipes and cleaned the data, adding nutritional values from the USDA API and estimated ingredient costs using ChatGPT. The final dataset includes fields like Name, Ingredients, Total Protein, Total Fat, Total Carbs, Total Calories, and Estimated Cost.
The dataset is diverse in recipe types, supporting meal variety. Preprocessing focused on standardizing ingredient names and units, ensuring consistency across recipes. Class distribution is generally balanced, though we might think of adjusting some overrepresented categories, like pasta and rice dishes, to avoid bias, or select specific recipes to train our data and generate good meal planning menus.
Challenges include the lack of serving sizes, limiting precision in nutrition and cost estimates, and the absence of preparation times, making time-based optimizations difficult.

Baseline

The baseline approach evaluates each recipe by calculating a health score based on macronutrient balance and a cost score that considers the nutritional value per dollar.

0.1 Health Score Calculation

Each recipe's health score is derived from its macronutrient balance, targeting an ideal ratio of 30% protein, 30% fat, and 40% carbohydrates (typical guidelines for a balanced diet). The score emphasizes protein content (weighted more heavily) and penalizes deviations from the ideal fat and carb ratios:

- **Protein Ratio**: Calculated as the proportion of calories from protein, with higher values boosting the score.
- **Fat and Carb Ratios**: Penalized based on their deviation from the 30% (fat) and 40% (carb) targets.

The final health score is computed as follows:

$$\text{health\_score} = (\text{protein\_ratio} \times 2.0) - |\text{fat\_ratio} - 0.3| - |\text{carb\_ratio} - 0.4| \tag{1}$$

0.2 Cost Efficiency Calculation

To balance health and cost, we calculate a `combined_score` that considers both the health score and the cost efficiency. Recipes are rated based on the cost per calorie, with a preference for recipes that deliver more nutrients and calories per dollar.
The weighted score formula uses weights w_1 for health and w_2 for cost efficiency:

$$\text{combined\_score} = (w_1 \times \text{health\_score}) - w_2 \times \left(\frac{\text{Estimated\_Cost}}{\text{Total\_Calories}}\right) \tag{2}$$

Here, we set $w_1 = 0.7$ and $w_2 = 0.3$, giving more emphasis to health but factoring in cost efficiency.

0.3 Recipe Selection

We randomly select recipes until reaching the budget limit ($50) or the desired number of recipes (5), ensuring the final selection fits within the constraints. The selected recipes are then displayed along with their protein, calorie, and cost values, providing a quick overview of their nutritional and cost impact.

```
# Function to calculate health score
def calculate_health_score(row):
    protein_ratio = row['Total_Protein'] / row['Total_Calories'] if row['Total_Calories'] > 0 else 0
    fat_ratio = row['Total_Fat'] / row['Total_Calories'] if row['Total_Calories'] > 0 else 0
    carb_ratio = row['Total_Carbs'] / row['Total_Calories'] if row['Total_Calories'] > 0 else 0

    health_score = (
        protein_ratio * 2.0 -
        abs(fat_ratio - 0.3) -
        abs(carb_ratio - 0.4)
    )
    return health_score

# Weights for scoring
w1 = 0.7  # Weight for health score
w2 = 0.3  # Weight for cost

# Calculate scores
recipes_df['health_score'] = recipes_df.apply(calculate_health_score, axis=1)
recipes_df['combined_score'] = (w1 * recipes_df['health_score'] -
                              w2 * (recipes_df['Estimated_Cost'] / recipes_df['Total_Calories']))
```

```
# Random baseline function
def random_baseline(recipes_df, budget, num_recipes):
    selected_recipes = []
    total_cost = 0

    while len(selected_recipes) < num_recipes and total_cost <= budget:
        recipe = recipes_df.sample().iloc[0]
        if recipe['Name'] not in [r['Name'] for r in selected_recipes]:
            if total_cost + recipe['Estimated_Cost'] <= budget:
                selected_recipes.append(recipe)
                total_cost += recipe['Estimated_Cost']

    return selected_recipes, total_cost

# Example usage
budget = 50
num_recipes = 5
selected_recipes, total_cost = random_baseline(recipes_df, budget, num_recipes)

# Save the selected recipes to a CSV file
selected_recipes_df.to_csv('baseline_selected_recipes.csv', index=False)
```

0.4 Evaluation

This baseline offers a basic comparison point, using a mix of health and cost scores. This method helps establish a minimum level of nutritional and budget optimization, which can be used to measure improvements in future, more advanced selection algorithms.

Main Approach

Our approach centers on an optimization algorithm designed to generate weekly meal plans that satisfy health and budget constraints. The algorithm scores recipes based on nutrient and cost metrics, while minimizing ingredient redundancy. User inputs include:

- **Budget**: Total dollar amount allowed.
- **Number of People**: Individuals to serve.
- **Recipe Count**: Number of unique recipes needed for the week (e.g., 14 for two meals per day).

The model outputs a menu that optimizes for nutrition and cost-effectiveness, adjusting based on individual dietary and financial requirements.

Algorithm Details

Our scoring function balances health and cost:

- **Nutrient Scores**: Recipes are evaluated based on macronutrient ratios.
- **Cost Efficiency**: Cost per calorie is factored to prioritize nutrient density within budget.
- **Variety Factor**: Ingredient redundancy is minimized by tracking ingredients across selected recipes.

The objective function maximizes:

$$\text{combined\_score} = w_1 \cdot \text{health\_score} - w_2 \cdot \left(\frac{\text{Estimated\_Cost}}{\text{Total\_Calories}} \right) - w_3 \cdot \text{redundancy\_penalty}$$

where w_1, w_2, and w_3 are tunable weights for health, cost, and variety.

Selection Process

The algorithm iteratively adds recipes to the weekly menu:

1. Calculate initial scores for each recipe.
2. Select the recipe with the highest combined score.
3. Update ingredient frequencies to adjust redundancy penalties.
4. Repeat until reaching budget or recipe count.

Example Scenario

For a user requesting a $50 meal plan:

- The algorithm selects a high-protein, low-cost recipe (e.g., a bean dish) to maximize protein density.
- Scores are recalculated, penalizing recipes with redundant ingredients if already used.
- The process continues until all criteria (budget, variety, recipe count) are satisfied.

Following discussions with the project mentor, the algorithm may be enhanced by allowing the user to specify a weekly calorie limit or custom macronutrient distribution. This would refine the scoring function for more personalized dietary goals.
This approach provides a balanced meal plan tailored to cost, health, and variety.

Evaluation Metrics

Our evaluation will use both quantitative and qualitative metrics:

- **Cost Efficiency**: The percentage of the user's budget utilized in weekly meal plans.
- **Nutritional Balance**: Assessed by scoring protein, carbohydrate, fat, and vitamin adequacy against daily recommended values.
- **Recipe Variety**: Measured by the number of unique ingredients in a weekly plan.

Qualitative metrics include personal satisfaction ratings and assessment of whether the generated meal plans are practical for actual use.

Results and Analysis

Baseline performance meets quantitative metrics but falls short on qualitative aspects. The generated recipes lack nutritional balance and appeal for a week-long meal plan.
One challenge is that the initial dataset lacks serving size information, complicating cost and nutrient calculations per person. Additionally, the dataset includes items such as sautéed carrots or pasta sauce, which may not qualify as complete meals. To address this, as discussed in the Future Work section, the next step will involve refining the dataset by selecting recipes that meet the criteria of complete, balanced meals.

Future Work

Our first priority will be to improve the dataset by selecting recipes that are suitable as full meals and specifying the intended number of servings per recipe.
Future steps include refining the scoring model, enabling user-specific nutritional preferences, and potentially implementing collaborative filtering to accommodate individual taste preferences. We also plan to explore partnerships with grocery delivery services to provide real-time ingredient price updates, enhancing cost accuracy. Additionally, we will investigate machine learning-based clustering for ingredient substitutions, which could allow users to input dietary restrictions or ingredient preferences.

References

[1] Adrian Jenssen L. Pe, Jerahmeel Kua Coching, Seth Gabriel D. Yeung, Wynnezel Akeboshi, and Robert Kerwin C. Billones. Cost optimization for weekly meal planning of college students based on calorie constraints using linear programming (lp) method. In *2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, pages 1–6, 2022.

[2] Jessica K. Wallingford and William A. Masters. Least-cost diets to teach optimization and consumer behavior, with applications to health equity, poverty measurement and international development. *arXiv preprint arXiv:2312.11767*, 2023.