Madelyn Yarber & Xavier Diaz

Professor Mithila

CIS4930-0008

17 November 2024

**CIS4930 Project Report - Group 2**

We started our project by creating the ingredients, preferences, and recipes files. The “ingredients.csv” file is organized by ingredient name, quantity, and unit. The user manually writes the ingredients they have into this file. Our project contains a “preferences.json” file that contains the diets: omnivore, vegetarian, vegan, and pescatarian. Each diet has it’s own set of nutritional goals. For recipes, we created a json file that contains the recipe name, the type of diet, the ingredients and their quantities, and the nutritional information of that recipe. All three of these files are imported into our program.

Within the file “main\_menu.py” we have a user interactive menu to allow the user to select their dietary preference. The chosen preference goes into our recommend\_recipes function that isolates the recipes that correlate with that preference, which ensures that the recipes match the user’s dietary preference. The recommend\_recipes also sorts the recipes by the number of ingredients as a way to order and make it easier for the user to view and pick recipes. After recommending recipes, the user is asked to select which recipes thay want to remove or keep for the meal plan within the pick\_from\_sorted function. The user can select certain meals and type “done” or type “all” to chose all remaining recipes. All remaining recipes will be random selected to be in the meal plan.

For the actual meal plan generation, which takes place in function create\_meal\_plan, recipes are randomly assigned to each day of the week. The algorithm is set up so that the total calories for each day are within 200 calories under and/or over the target calorie range. The target calorie range is defined in the preferences.json file depending on dietary preference. The resulting meal plan is then written onto a file names “meal\_plan.txt”. After creating the meal plan, the program then calculates what ingredients are missing based on the user’s available ingredients in the “ingredients.csv” file. The algorithm takes the total quantity of ingredients used in each recipe and subtracts that from the quantity of any ingredients available in the csv file. These ingredients needed are then written to a shopping list appended to “meal\_plan.txt”. Lastly, the algorithm used matplotlib to created a stacked bar chart showing the nutrional totals for calories, protein, carbs, fat, and fiber for each day. This provides a visual representation of the meal plan’s nutrional balance for that week.

One reason we chose this algorithm was for its personalization and user interaction. Our algorithm is set up so that it allows the user to chose their dietary preference, remove recipes, and add recipes to their meal plan; an important feature is that only the recipes that match the user’s dietary preference are included. The algorithm is also flexible and balanced in the way that the calories than be 200 above or below the daily calorie goal, as it ensures that users don’t have an overly restrictive meal plan.