MealCheck: A Real-Time, Adaptive and Interactive Meal Planner

Pranav Patil (34790252)
Purdue University
West Lafayette, Indiana, USA
patil127@purdue.edu

Raj Jagtap (34781579)
Purdue University
West Lafayette, Indiana, USA
rjagtap@purdue.edu

Rucha Deshpande (34780087)
Purdue University
West Lafayette, Indiana, USA
deshpa37@purdue.edu

ABSTRACT

With increasing health consciousness among people, the importance of meal planning is growing day by day. People prefer enjoying a variety of food items while maintaining a balance in their nutrition. In today's fast-paced lives, it has become increasingly difficult to follow a balanced diet plan due to frequent changes in daily schedules. There are a variety of meal planning applications in the market that offer meal suggestions based on the user's nutritional needs and food preferences. But these apps do not adapt to the frequent changes in users' schedules and food habits. We propose a mobile application that adjusts subsequent meals in real-time in accordance with changes by the user as a result of unplanned food intake, thereby meeting the daily nutrient requirement. We place a strong emphasis on user engagement to learn about the user's culinary tastes from his past selections and train our model to make recipe suggestions based on them. We plan to evaluate the model by performing two user studies and a system scalability test.

CCS CONCEPTS

• Computing methodologies → Artificial intelligence; **• Human-centered computing → Human computer interaction (HCI)**.

KEYWORDS

Recommendation System, Mobile Application, Optimization, Sequence Models, Meal Planning

ACM Reference Format:
Pranav Patil (34790252), Raj Jagtap (34781579), and Rucha Deshpande (34780087). 2022. MealCheck: A Real-Time, Adaptive and Interactive Meal Planner. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Increased consumption of highly processed foods and changing lifestyles have contributed to people eating unhealthy diets high in calories, fats, free sugars, and salt[12]. Growing fast food chains and quick on the go meal options have led youngsters between the age of 18 to 29 often face a challenge to eat healthy[8]. People frequently rely on fast food to get through the day without thinking about the health risks associated with it. It is advisable for people to plan their meals ahead of time to maintain a healthy diet[9]. In the long run, meal planning can help save time, and ensure more balanced healthy meals.

As per a survey, 64% of people in the age group of 18-29 regularly use or have used a diet planner app [8]. These apps can create a diet plan based on user's weight, height, dietary preferences health goals, etc.[4] They work on the Input-Output model i.e. after taking input of user data and preferences, apps suggest a weekly or daily meal plan which balances nutrients such as proteins, vitamins and carbohydrates. Existing meal planning apps focus on user's nutrient consumption but do not take into account everyday routine changes (unplanned changes in a user's calorie intake)[3]. Consider a scenario where a user has received a meal plan from an app but he could not follow lunch as specified by the planner. Shall he still follow same dinner as suggested? To balance his diet, the dinner suggestion should change as per his day's food intake. The apps that are currently available do not change meal plans in real-time[4][3]. In this case, the user either does not change the dinner food or changes dinner based on approximate knowledge of diet requirement. These unmanaged changes on balanced diet can cumulatively affect user's desired results adversely[5].

A get-around followed by some apps to get over this unbalance in diet is, the apps asks users to write what they ate in whole day and show which nutrients are required to be eaten on that day for a good daily diet. However, people don't eat nutrients, they eat food. The people who strictly follow a curated meal plan usually cook meal at home. As per a survey [13], 40% of Americans faced challenges in having to plan different meals on a daily basis. Currently, there does not exist an app that can create a meal plan and can accommodate changes in the day's food intake and then suggest a next meal to cook.

The recipe recommendations by existing apps are based on user preferences for which they require detailed user inputs. These diets are generally prescribed for an entire week and tend to become restrictive as a user's eating preferences may change based on their mood, weather or other factors. Studies show that fatigue and time scarcity decreases the likelihood of following meal plans[14]. Existing apps do not ensure a user's adherence to the meal plan as user tends to neglect suggested meals due to factors like time limitations or undesirable suggestions.

We propose MealCheck an easy-to-use android application that takes that suggests recipes based on users' preferences and adaptively takes into consideration any changes in their eating schedule to change the recommendations given to the user in real-time. The app also gives users a meal plan based on their nutritional requirements determined by their height, weight, age, gender, and target weight. We propose a simple method to find recipes that are close to the user's preferences and yet far from the user's recent consumption to allow users variety in their meals. The user can select

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

how similar they want their next meal to be compared to the recent meals they have had to change the recommendations. The app also allows the user to have a different meal from the meal plan and then add the meal in the app to change the recommendations so that the nutritional requirements are still satisfied.

We evaluate this system using by the means of a study involving 12 participants. The users filled out a questionnaire to answer questions regarding the usability and functionalities of the system and suggest required improvements. The results of the survey showed that the overall user attitude towards the app was positive. The system performed satisfactorily in maximizing variety while adapting to changes in the previous meal intakes.

2 RELATED WORK

Our work focuses on providing a nutritious meal keeping in mind the preferences set by user and variety asked by user. Lifesum App [3] provides a 7-day meal plan with tasty nutritious recipes. It also provides an option to customize the meal plan to fit user's preferences. However, this meal plan is locked in at the beginning of the course. The plan set by the app fails to accommodate user's behaviour such as cheat meals, etc. An app that can adjust future meals considering the impact of the meals already taken by user should be a better and easier fit for users with less time to focus on meals.

Healthify.me [4] app solves the problem of adapting meals to what user has already eaten by collecting a list of items eaten by user. This app, however, works by only suggesting amount of nutrients required by the user to achieve daily goal. However, user is not suggested a recipe that might help in achieving that goal. A user is required to look at the suggested nutrient values and find a recipe on their own either using the database of the app or the internet. This can be a problematic task for users that are new to cooking. Users may have repetitive dishes if there is no better recommender system available that can suggest a variety of meals as compared to past recipes eaten.

A study has presented "Plan-Cook-Eat" [6], a web application that makes meal plans in accordance with the essential macronutrient distribution of daily calories based on people's Total Daily Energy Expenditure (TDEE). The default ratio for the distribution of macronutrients is based on the Acceptable Macronutrient Distribution Range (AMDR) developed by FNRI-DOST [1] in the Philippine Dietary Reference Intakes. However, app users could still set their own preferred macro ratio to meet their unique nutrient intake requirements. To guarantee that the total number of calories is accurate, each type of macronutrient (protein, carbs, and fat) is taken into account. The amount of food for each meal plan is split by the number of meal intervals that users specify. No matter how many times a user wants to eat throughout the day, their TDEE still limits their daily caloric intake. However, this app's restriction is that users can only change their calorific intake to suit their preferences. The recommended meals do not account for any user preferences or dietary habits.

A method to suggest recipes based on previous recipes made by user is proposed by Majumder et.al. [11]. Their work focuses on suggesting new recipes by mixing the previous recipes. This is a smart way to add variety to the meals suggested as the new recipes created are always little bit different from the previously made recipes. However, the shortcomings of this approach are the recipes created are similar to previous recipes in larger sense. A user that has only explored recipes with rice may not be ever suggested with a recipes that includes tortillas. We propose a new approach that will suggest a recipe that is completely different from the meals eaten by user recently but keeping in mind the preferences set by the user. Our work uses the dataset published by their team.

Another study [15] with the same objective focussed on recommending healthy meal plans by optimising nature-inspired many-objective diet problem. The diet problem was modeled as a multi-dimensional, many-objective knapsack problem. The goal was to choose a portion of a set of food products so that all objectives are achieved concurrently while staying within the limits of a knapsack. The study used only cost, preference, and preparation time as objectives. The dietary problem was formulated to minimize the value of the cost and handling/preparation time of the food while maximizing preferences. This was subject to the upper and lower limits of nutrient requirements which was the boundary constraint. The optimization problem was solved using Non-dominated Sorting Genetic Algorithm III [7]. We propose an easier method that is more deterministic and tractable as compared to the genetic algorithm.

3 APPROACH

We build an easy-to-use mobile application that gives users a week-long diet plan, considering a recent past intake in terms of nutrients, eating habits, and patterns. We opt for a mobile application considering its handiness over websites. In the app, upon registration, users are asked about height, weight, age, weight goal, etc. Using this information an optimal nutrition intake is calculated. For setting initial dietary preferences, users are asked to select a minimum of 10 of their favorite or preferred dishes. Using these dishes, the user's diet goal and nutritional requirement a curated week-long diet plan is build for the user. The users are able to log their food intake everytime after a meal.

The system is designed to be adaptive to unexpected changes in the user's nutritional consumption by virtue of a cheat meal or unintentional changes in the schedule. If the user is not able to follow the diet plan perfectly, then the app suggests a new meal to compensate for the excess or lack of nutritional intake. To accomplish this, the app asks the user to enter the food that he replaced the suggested meal with. The users get alternative suggestion based on their current consumption as well as their preferences built over time.

3.1 Proposed System

To achieve adaptiveness in the meal suggestion, we propose a novel approach to recommend the next meal to cook based on the nutritional values of previous meals users has had and user preferences also maximizing variety of meals. We build a feature vector space by encoding the recipes from the Food.com recipes dataset [11] using Doc2Vec [10] method. This method converts each of the text recipe into a 100-dimensional vector. Let us call this vector space $\mathcal{R}$. Doc2Vec captures the order and context of words and semantics of a paragraph in generating vector embeddings. This helps us encode the recipes such that the recipes that are similar in terms

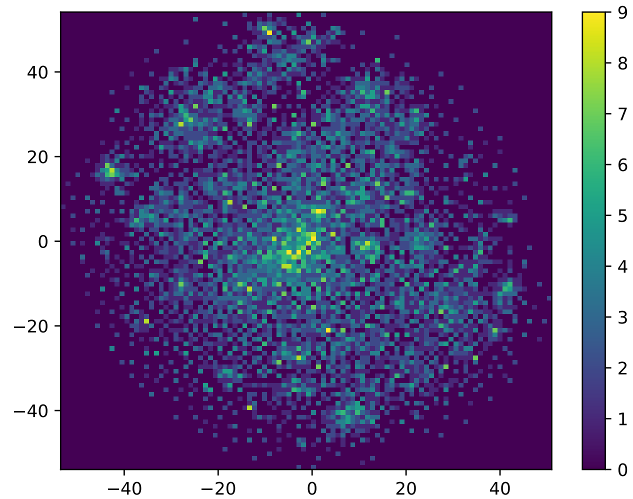

Figure 1: Heatmap of the t-SNE of $\mathcal{R}$ Vector Space

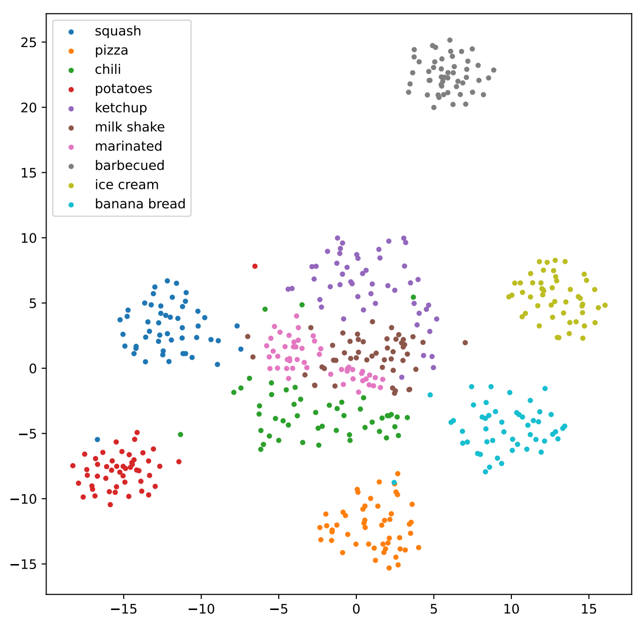

Figure 2: Scatter Plot of 10 Clusters after Sub-sampling of $\mathcal{R}$ Vector Space

of ingredients and cooking steps are close together in vector space. To test this, we convert 100-dimensional vectors to 2D using t-SNE [16] and plot them. Figure 1 shows the heatmap of t-SNE of $\mathcal{R}$ vector space. We can observe clusters in the figure which mean that similar recipes are clustered together. The clusters can be observed clearly in Figure 2. In this figure, we sub-sample only 50 recipes each of some specific types. We can observe that different types of recipes are part of their clusters as expected. Doc2Vec model has two components namely word vectors and document vectors. Using word vectors we can observe which words are syntactically closer to each other. Here we show similar words and their similarity to *pasta* as ranked by the model:

(1) *macaroni*: 0.87
(2) *spaghetti*: 0.83
(3) *tortellini*: 0.82

Also, the document vector space was able to capture similar recipes. Here we show similar recipes to a recipe document for *Sausage Pizza* as ranked by the model:

(1) *Sausage Pizza with Cheese*
(2) *Sausage Crescent Roll*

This recipe vector space $\mathcal{R}$ allows us to make a mathematical similarity metric to capture preferences of the user and suggest varieties in the next meal.

We build a feature vector space based on Nutritional Value of ingredients. Let us call this vector space $\mathcal{NV}$. Here, each dimension represents a value of a nutrient such as protein, vitamin A, fats, etc. Each recipe can be represented in this vector space as a point.

3.2 Methodology

Since each user is to be shown a recipe that is away from the recipes that they had in recent days but closer to user's preferences that are built over a longer duration we encode a user vector r_{user} in the $\mathcal{R}$ vector space. This r_{user} is a linear combination of recipes they had in previous days. r_{user} consists of two components $r_{user,p}$ and $r_{user,c}$.

$r_{user,p}$ is the user preference vector. It is used to identify user's comfort zone in meals. This vector is the weighted sum of all the recipes user has had where the weights are proportional to the ratings user has given to the recipe. Thus the value of the component is given by:

- During Initialization:

$$r_{user,p}^{(i+1)} = \frac{1}{i}\sum_{k=0}^{i} r_k$$

- Updating after user tries a recipe:

$$r_{user,p}^{(i+1)} = \frac{1}{i+1}\left(i \cdot r_{user,p}^{(i)} + w_k \cdot r_k\right)$$

$r_{user,c}$ signifies the user's current consumption and is used to identify types of recipes user has tried recently. Since recent recipes are to be given more importance, we use exponential decay for the $r_{user,c}$ to reduce the importance of the previous meals by a decay factor β and add the suggested recipe (r_i) corresponding to the user rating. This component is thus adjusted as follows:

$$r_{user,c}^{i+1} = \beta r_i + (1-\beta) r_{user,c}^{i}$$

Ideally we need suggestions to be different from the recent consumption while taking into account the preferences that are built over time. To achieve this we define two spheres S_p and S_c with centers $r_{user,p}$ and $r_{user,c}$ respectively (refer Figure 3). The user can adjust the radius of the sphere S_c (λ) proportional to the variety they desire. The sphere S_p is a larger sphere that envelopes the sphere S_c such that they touch at a single point. The radius of sphere S_p is calculated as the sum of λ and distance between $r_{user,p}$ and $r_{user,c}$. To ensure that the variety is maximized while taking

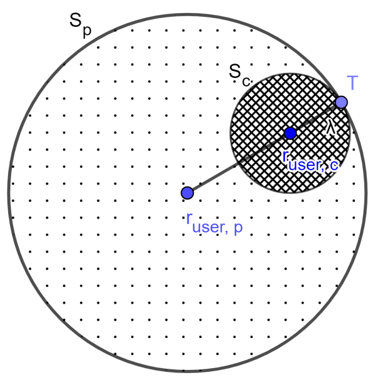

Figure 3: Spheres S_p and S_c used to Find the Desired Area (Represented by Dots)

into consideration the user's preferences, we suggest the recipes that lie outside sphere S_c but inside S_p.

This can be explained using equations as follows:

$$R_{\text{valid}} = \{r_i | r_i \in S_p\}$$
$$R_{\text{invalid}} = \{r_i | r_i \in S_c\}$$
$$R_{\text{shown}} = R_{\text{valid}} - R_{\text{invalid}}$$

Using the above equation we find set of recipes that can be suggested to the user to maintain variety as well as preference. Suppose a meal should have nutrients with nutritional value nv (vector in $\mathcal{NV}$ vector space). We calculate nv by dividing the total nutrituional requirements into equal portions and then taking into considieration the day's previous meals' deviation from nutritional requirements. For example, first meal's nutritional value deviates from required nutritional value nv by Δ then for the next meal the updated required nutritional value will be $nv = (nv_{\text{original}} - \Delta)$.

We reset the nutritional requirements everyday i.e. for first meal of the day.

To suggest a recipe, we find the difference δ_r between nutritional value of all recipes r (nv_r) in R_{shown} and nv. We then filter the recipes based on absolute value of δ_r that satisfy a threshold η ($|\delta_r| < \eta$).

The dataset contains $\approx 230k$ recipes. Computing distance between vectors and filtering in every request to get suggestions would be computational expensive and will slow down the system. To reduce the number of comparisons and improve the efficiency of the algorithm, we employ K-means to find the clusters and their centroids. The number of clusters used in this optimization is 10000. We use these centroids as a proxy for the recipes in the cluster. In the first level of filtering, we find the clusters that satisfy the requirement and use the recipes in those clusters thus reducing the number of computations by a significant margin.

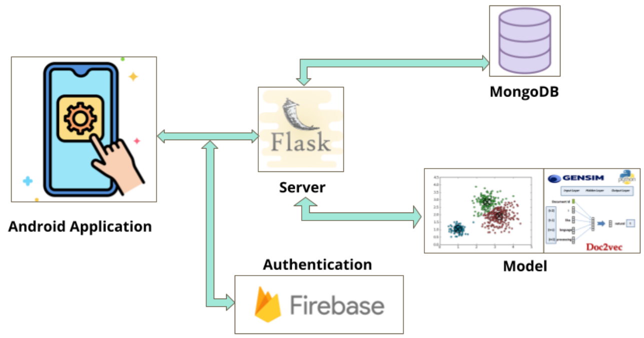

Figure 4: System Architecture

3.3 Meal Plan Generation

The app generates a week-long diet plan based on the past preferences of the user. For generating each meal suggestion in the diet plan, we use a simulated r_{user} that is updated in accordance with the previously included meals in the plan. We assume that the user accepted the latest generated suggestion in the meal plan, and simulate future r_{user} accordingly. After the generation of this week-long meal plan, r_{user} will be reset to its current position.

3.4 System Architecture

Figure 4 shows the architecture of our system. The system is developed in Python-3 using Flask framework for the back end and deployed on DigitalOcean Droplet http://159.223.103.200:2080. The android application is developed in Java with user authentication using Firebase. The source code for the Flask Server and Android Application is available on GitHub[1]. A docker-support branch is also created with code to deploy the server using Docker.

3.5 User Interface

The user interface is designed to be minimalistic and simple so that any user from different age groups can use the app with ease without assistance. The app is divided into components such that the user doesn't have to remember lot of things and can rely on recognition rather than recall thus reducing short-term memory load.

3.5.1 Sign up and On-boarding (5).

The user can create an account to sign into the app using their email and password. Once the user creates their account, they are prompted to add their details such as name, height, weight, target weight (the weight that the users aim to achieve), and gender. This helps us to keep track of the user features that are necessary to calculate the Basal Metabolic Rate or BMR which gives the amount of energy expended per day at rest. BMR gives the approximate daily calorie requirement for a person. BMR is calculated using the revised Harris-Benedict Formula [2] as follows:

$$BMR_{male} = 13.397W + 4.799H - 5.677A + 88.362$$

$$BMR_{female} = 9.247W + 3.098H - 4.330A + 447.593$$

[1] https://github.com/jagtapraj123/MealCheck

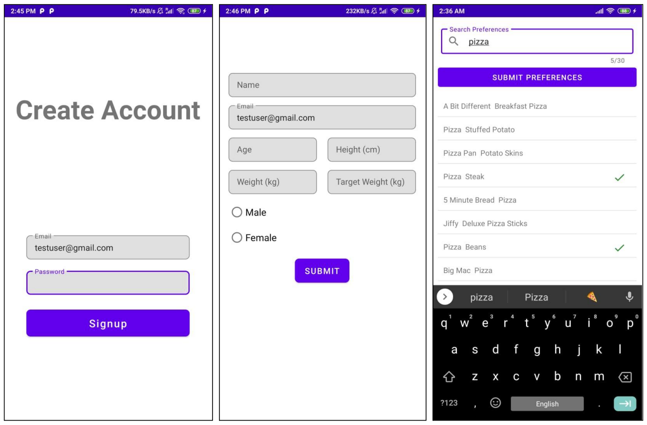

Figure 5: Onboarding Workflow

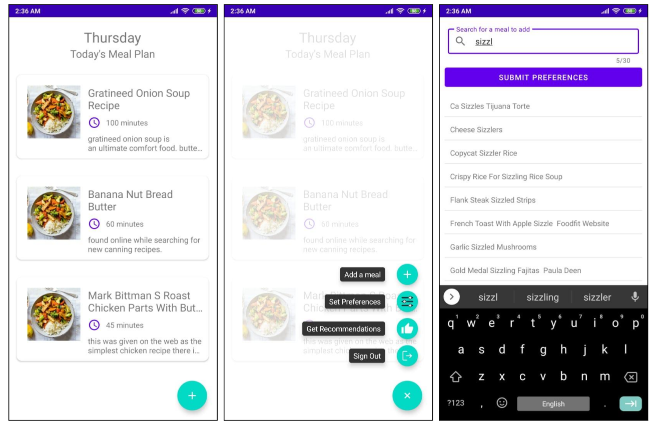

Figure 6: Left to right (a) Home Screen with day's meal plan (b) Options to add a meal, set preferences, get recommendations, sign out, (c) Add meal screen where user can search for the meal they had.

where:
W is body weight in kg
H is body height in cm
A is age
F is body fat in percentage

Once the user data is taken, the user is prompted to select at least 10 of their favorite recipes. This is necessary to build the user's preferences vector $r_{user,p}$. The user can search for their favorite recipe and select it before proceeding to submit the preferences. To allow easy reversal of the action, the user is allowed to reset the preferences and select new recipes as favorites from the homepage.

3.5.2 *Home Screen (6).*

The user is taken to the homepage once they set their preferences. Using these preferences we initialize $r_{user,p}$ as a linear combination of the preferred recipes selected. $r_{user,c}$ is initialized randomly at a certain distance from $r_{user,p}$. The server then simulates the user behavior by suggesting a recipe and then considering that the user like that recommendation, and suggests the next meal. We ensure that the daily nutritional requirements are met by combining the intake through the three suggested meals for the day. Similarly, a plan is generated for the week and sent to the user.

After the generation of the meal plan, the user can see the day's plan on the home screen. Additionally, they have options to add a new meal, set preferences, get recommendations, and sign out.

3.5.3 *Add Meal.*

If the user willingly or unwillingly due to some reason don't follow the meal plan for the day, they can add a meal that they have had and it will be considered instead of the recommended meal in the plan. Users can search from all the recipes and select the one they had. This change in the schedule is taken into consideration and when the user asks for recommendations for the next meal, they have suggested recipes that account for the change in the meal.

3.5.4 *Get Recommendations (7).*

The user can ask for custom recommendations if they are not satisfied with the meal plan or want to try something different. The user

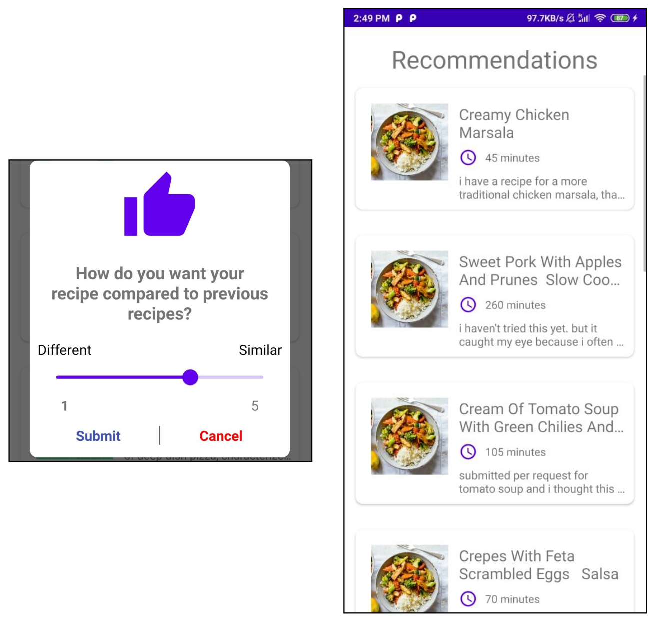

Figure 7: Left to right (a) Dialog to get the user's choice of similarity of the dish, (b) Recommendations according to user's input

is prompted to select how they want their meal, similar to the previous meals they have had or they want to try something different. This is used to set λ. We also scale λ such that $2.5 \leq \lambda \leq 3.5$. Meals are generated using the method described in 3.2 The user gets a list of recipes that match their preferences, recent consumption, and similarity level that they set.

3.5.5 *Recipe Details and Rating (8).*

Whenever a recipe is suggested to the user, they can click on the recipe to see the details. The details include the recipe description,

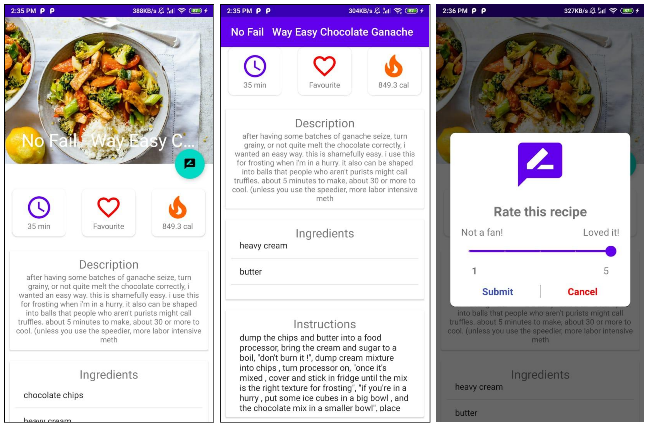

Figure 8: Left to right (a) Dialog to get the user's choice of similarity of the dish, (b) Recommendations according to user's input

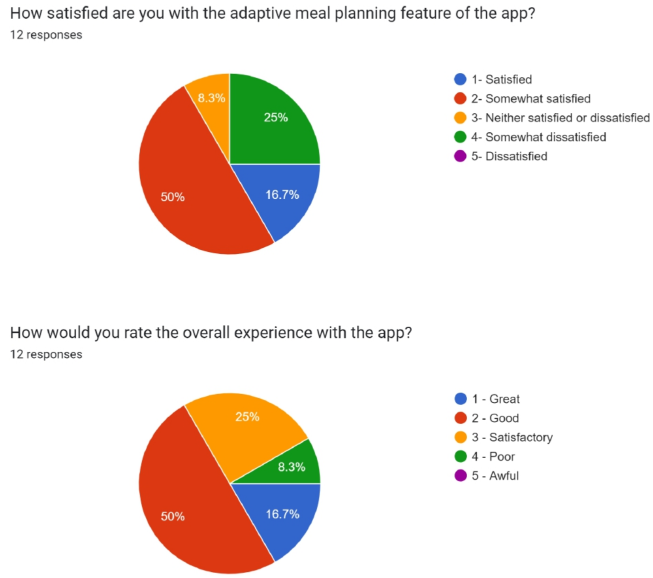

Figure 9: (a)User responses about adaptive meal planning feature, (b)User responses regrading overall experience with the app

preparation time, nutritional value, the ingredients used in the recipe, and the steps to cook. The user can also rate the recommendation using the rating button which prompts a dialog to ask the user for a rating on a scale of 1 to 5.

4 EVALUATION

To evaluate the app and the model we performed a user study with 12 participants. The participants were our friends between the age group $20 - 30$. Through this study, we evaluated the proposed system in terms of:

(1) Usability of the app to a new user.
(2) User agreement with the recipe suggestions provided by the model.

In the usability study, users who have not interacted with the app before were given 15 minutes to setup their account such as completing signup, setting preferences, etc. After setup, the app provides a week-long diet plan to each user based on preferences entered. The users were then allowed to interact with the app and simulate one-day interaction with the app. This included generating recipe suggestions and adding meals that they had during the hypothetical day.

In the user agreement study, users were given a 30-minute tutorial on how to use the app correctly and explained how the model works on a basic level. In the setup step, users were asked to simulate one week of interactions to get over the problem of a cold start during the experiment. After this, in the actual study, users were asked to simulate one more week of interactions with the app. After every recipe suggestion, users were asked to rate their suggested recipe.

After using the app, the users were given a questionnaire to fill out. The questionnaire contained questions regarding the usability of the app, the relevance of the suggestions in terms of nutrition, variety and user preferences, and their experience with meal planning apps. The users were asked to give suggestions for improvement in the app.

Overall, the users were satisfied with the adaptive meal-planning feature of the app. 75% of users were happy with the variety of recommended meals. 50% of users felt that the suggestions didn't perfectly fit their preferences. Also, only 50% of the participants found the suggested meals satisfactory in terms of nutrition. Other users had anticipated more healthy recipe recommendations. When asked how the suggested system compared to other apps they had used in the past, 50% of the participants claimed that our app was better. In the 'suggestions for improvement' section, one user suggested that the "setting preferences" feature could include a list of recipes to choose from rather than forcing the user to recall his favorite foods. Another user stated that the model did not adequately consider his food choices because it offered non-vegetarian meal suggestions despite of the user providing only vegetarian meal preferences. About 67% users were positive about the adaptive meal planning feature of the app. The overall response to the ease of use of the app was positive.

5 DISCUSSIONS

Existing meal planning apps do not adjust to unexpected changes in users' daily diets. We have suggested a system that offers meals while adjusting to dietary changes considering user preferences. The app is an interactive system where the users can set their preferences, log their meal intakes and get revised suggestions by adjusting the amount of similarity or variety they want in the next meal. The model aims to increase the variety of suggested meals by keeping track of the most recent meals the user has had.

In order to assess the usability and functionality of our software, we conducted a user study. Overall, the app received positive user reviews. Users expected the recipe recommendations to include more nutrient-dense meals. However, the crowdsourcing dataset we used was compiled from individual contributors. As a result, it is not just limited to meals with higher nutritional values. This suggestion would be better infused if a dataset of meals with higher nutritional values was available. Some users reported about the mismatch between their preferences and suggested meals, especially between vegetarian and nonvegetarian food choices. If the dataset contains a label indicating the type of meal, this discrepancy can be eliminated. Currently, we initialize the user consumption vector randomly at some distance from the preferences vector. As a result, the first few meal suggestions may include a wide range of meals, reducing the impact of user preference. The app balances meal portions throughout the day but does not currently distinguish between meal types like breakfast, lunch, and dinner. In the daily meal plan, future work may incorporate the user-specified meal proportion distribution. Other future work tasks would include a better approach for initializing the user consumption vector $r_{user,c}$ and tuning hyperparameters like λ.

6 CONCLUSION

In this project, we tried to develop an interactive and adaptive meal-planning app. This system is designed to incorporate the preferences of the user as well as suggest variety of meals so that user is able to try different meals. We were able to encode recipes in mathematical vector-space which gave the system the ability to compare recipes based on their ingredients and cooking procedure. We also performed experiments to evaluate the quality of these vector spaces.

During the project, we also conducted a user study that gave us insights into which features can be improved in the app as well as which features can be a good addition to the app.

In this project, we presented a smart, adaptive app that can be used for meal planning. With a more descriptive dataset, this app has the potential to be a valuable part of people's day-to-day meal planning.

REFERENCES

[1] [n. d.]. . http://enutrition.fnri.dost.gov.ph/site/uploads/2015_DIETARY_SURVEY.pdf

[2] [n. d.]. Calorie Calculator. https://www.calculator.net/calorie-calculator.html

[3] [n. d.]. *Lifesum Health App.* Retrieved September 23, 2022 from https://lifesum.com/

[4] [n. d.]. *Track Nutrition, Plan Your Diet, Get Fitter.* Retrieved September 23, 2022 from https://www.healthifyme.com/us/

[5] 2022. *15 Meal Planning Statistics For a Healthier 2022.* Retrieved September 22, 2022 from https://www.soocial.com/meal-planning-statistics/

[6] Manuel B. Garcia. 2019. Plan-Cook-Eat: A Meal Planner App with Optimal Macronutrient Distribution of Calories Based on Personal Total Daily Energy Expenditure. In *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM).* 1–5. https://doi.org/10.1109/HNICEM48295.2019.9073490

[7] Qinghua Gu, Rui Wang, Haiyan Xie, Xuexian Li, Song Jiang, and Naixue Xiong. 2021. Modified non-dominated sorting genetic algorithm III with fine final level selection. , 4236–4269 pages. https://doi.org/10.1007/s10489-020-02053-z

[8] Alexander Kunst. 2019. *U.S. adults that would use an app to track diet and nutrition 2017, by age.* Retrieved September 23, 2022 from https://www.statista.com/statistics/698919/us-adults-that-would-use-an-app-to-track-their-diet-by-age/

[9] Alexander Kunst. 2019. *Usage of Health Apps to Track Nutrition among US Adults by Age 2017.* Retrieved September 23, 2022 from https://www.cdc.gov/healthyweight/healthy_eating/meals.html

[10] Quoc V. Le and Tomás Mikolov. 2014. Distributed Representations of Sentences and Documents. *CoRR* abs/1405.4053 (2014). arXiv:1405.4053 http://arxiv.org/abs/1405.4053

[11] Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating Personalized Recipes from Historical User Preferences. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).* Association for Computational Linguistics, Hong Kong, China, 5976–5982. https://doi.org/10.18653/v1/D19-1613

[12] World Health Organization. 2019. *Healthy diet.* Retrieved September 23, 2022 from https://www.who.int/health-topics/healthy-diet#tab=tab1

[13] Russell Redman. 2020. *Pandemic sways most Americans to eat at home more often.* Retrieved September 22, 2022 from https://www.supermarketnews.com/consumer-trends/pandemic-sways-most-americans-eat-home-more-often

[14] Amy Storfer-Isser and Dara Musher-Eizenman. 2013. Measuring parent time scarcity and fatigue as barriers to meal planning and preparation: quantitative scale development. *Journal of nutrition education and behavior* 45, 2 (2013), 176–182.

[15] Cumali Türkmenoğlu, A Uyar, and Berna Kiraz. 2021. Recommending healthy meal plans by optimising nature-inspired many-objective diet problem. *Health Informatics Journal* 27 (01 2021), 146045822097671. https://doi.org/10.1177/1460458220976719

[16] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html