NutriChoice

Elizabeth Binkina, Simon Hu, Nynika Badam

Project Summary

NutriChoice is a web application that allows for a user to compile a list of food ingredients to buy at the grocery store by inputting dietary goals and preferences. Rather than taking the extra time to research the number of calories in food items or trying to figure out how to increase your protein intake, this application will do it immediately, saving time.

Description

Being healthy and taking care of our bodies can be a hassle and something that takes discipline and dedication. However, many people tend to neglect their health because it's too much work or because it's just not their problem right now. So, in order to simplify the process, NutriChoice can be used to quickly input what kind of health goal you're trying to reach and then compile a list of ingredients that you can use to make meals with. Not only does this allow a person to maintain a healthy lifestyle, but also removes the hassle of figuring out what to buy.

A creative, technically challenging feature that could be implemented is generating grocery run reminders and meal plan reminders based on a user's generated grocery list for that week. A user would receive daily meal plan reminders consisting of the ingredients that should be eaten that day or throughout that week. When a user generates a list of ingredients, they will receive reminders to go to the grocery store to purchase these items until they mark that they have done it. These reminders can be generated through third party notification services like Twilio that can send reminders via SMS texts. The actual scheduling can be done by another service like Django which can be integrated into the web application.

Usefulness

Our application is useful because it gives the user a compiled list of ingredients to buy while also hitting their specified goals, whether it be weight loss or simply creating a healthier lifestyle. For example, a user was told by their doctor that they need to implement a diet with lower cholesterol. That user can use our web to find ingredients that have low cholesterol without spending an enormous amount of time searching random websites for what foods or ingredients to buy. On the web, consumers can input their specific criteria, whether it be higher protein, low sugar, low fat, etc. In return, they get a list of ingredients that better suit their needs. There are search engines that can tell you what foods have certain macros, but they will not tell you much more detailed statistics on that product.

Our dataset we plan to use has a wide and complex table of detailed statistics for each ingredient, such as vitamins, acids, oils, etc, which many other websites do not have. This level of detail

adheres to a wide majority of the population because people are unique and have all sorts of different goals and allergies when it comes to food. With our web, we can compile a relatively large list of ingredients with specified statistics where users can view the tradeoffs of eating certain foods. For example, say someone wants a low-fat ingredient list. In other search engines, they will find some foods that are classified as “low-fat”. But, what if they have an elevated amount of carbs, sugar, and are calorie dense which the user does not want? Our web would output low-fat ingredients that are also low-sugar, carb, etc with a high level of detail.

Realness

Our first dataset is a food_table that downloads in a xls file, but could also be converted to a csv. It is from data.gov website. It has 25 column variables and 2015 row items. All of these rows are food items and ingredients which have variables such as what kind of food it is (vegetable, dairy, oils, fat,etc) as well as calories and added sugars.

Our second dataset is also a list of composition of foods/ingredients but with high level detail. It was found on kaggle but its source is also data.gov. It is downloaded as a csv file. It has roughly 50 columns with more than 8600 items. All of these food items have a high number of details ranging from lipids, carbs, water, many types of vitamins and minerals,acids, and more. This dataset focuses more on the nitty gritty details of food rather than calories like the first dataset.

Functionality: PowerPoint slide mockup

Our final application will be a grocery list compiler that can help the user keep track of what ingredients they should buy to meet their nutrition goals. That means that our UI mockup will have a checkbox for labels such as “low cholesterol” or “high protein,” a search bar for specific ingredients and their nutritional information, and a slider for specific nutritional values. In addition, if the user wants to look at common dietary goals, then there will also be links to common dietary goals or plans to help the user find the right one for them. Once the user inputs their desired values, then our application will list several ingredients that match the description or values. The user can then add ingredients to a list that will then provide a summed nutrition count of that list. Lastly, the user can create an account to save their ingredients list or export the list as a PDF for future reference.

For an application like this, there are several steps that must be implemented. For example, datasets must be acquired and integrated into the application. Then the application must display all the information into user-friendly tables. Next, there are many features that must be implemented such as the search bar, nutrition value slider, checkboxes, user login, and shopping cart. Elizabeth will be responsible for integrating the datasets and external APIs into the application so we can display nutrition information for each grocery ingredient. Nynika will be responsible for creating user authentication and authorizations, exporting a list as a PDF, as well as creating the general UI design. Simon will be in

charge of creating tables for nutritional values, ingredient information, and features such as the search bar and checkboxes for the UI mockup. If bugs were to occur, everyone should offer a helping hand to one another to help resolve the issue through testing and debugging.