ETL PROJECT

Group: Andrew Bankston, Jacob Zacarias, Jonathan Hicks, Walaa Alani

Feb 24, 2021

Business Context:

We are starting a meal prep kit delivery service that is based on recipes from All Recipes. To start, we are targeting our service to people who have certain dietary restrictions as a way to make their meal-planning and meal-prep easier.

Our motto is *"Our food is delicious and nutritious so that you can live your life ambitious."*

We started our process in the healthy foods section of All Recipes (https://www.allrecipes.com/recipes/84/healthy-recipes/). Our goal was to find out which recipes customers are most likely to come across on All Recipes, and therefore most likely to want to try. As it can be difficult to prep meals for specific dietary restrictions, we decided to focus on recipes meeting those restrictions as a start. We also want to give customers the ability to browse our recipes and sort based on prep time, ratings, and nutritional value. We would also like to give customers the option to group recipes for a day's meals together to not exceed some selected nutrient maximum or group recipes based on common ingredients to make meal prep even easier (i.e., chop onion for 2 different recipes at once). To achieve these goals, we have to pull all the required information for each recipe: the recipe name, the prep time, the ingredients, a link to the recipe on All Recipes for the instructions, nutrition info, and an image to make the listing more enticing. Once these are loaded into a database, we would want to sum nutrient values, sort or select recipes by ingredients, rank recipes by prep time, rank recipes by ratings, and exclude recipes above a set nutrient max. Before finalizing the database for interaction with a customer-facing website, we would likely remove any recipes that have less than 3.5 stars and 50 reviews.

Extraction

We scraped our data from https://www.allrecipes.com/recipes/84/healthy-recipes/ and chose to pull data from different categories under the healthy recipes section. The categories for our data are as follows:

- Healthy Snack Recipes
- Low-Calorie Recipes
- Low-Carb Recipes
- Low-Cholesterol Recipes
- Low-Fat Recipes
- Weight-Loss Recipes

We accomplished this by first using BeautifulSoup to find all the <a> tags that correspond to the row of category icons under the banner image. From there, we pulled out the href links for the categories from our list above. We then iterated through the list of href links for our categories, and built a dictionary in which each category name is a key for a list of href links to recipes within that category. Finally, we iterated through that dictionary of recipe links and for each recipe, we extracted:

- the recipe title
- the recipe image
- the rating (out of 5 stars) and the number of ratings, both with try/except clauses in case there were no ratings
- the recipe prep time and total time
- the recipe ingredients
- the nutrient list

Transformation

Once we chose these categories to focus on, we conducted for loops to navigate through the website and derive certain information from each category. The information gathered consists of all staff picks and top recipes for each category, scraping the title, rating, number of ratings, prep time and total time needed to cook the meal, a list of ingredients, and the nutrition facts.

After extracting the links for the recipes, we saw that each recipe link was duplicated in the list. We removed duplicates by converting the list of recipe links to a set, which was then converted back to a list. These trimmed lists were then placed in a dictionary with the health category as their key. This allowed us to use the key as the collection name in our final step.

Once extracting data from each recipe, the cleaning process consisted of turning html into text. We were able to extract the title and image url without further clean up, but the other data required removing line break marks and spaces before and after the text we wanted. These were done using replace and strip methods. For ingredients, this involved a list comprehension through the ingredients list.

For time, data were extracted as a list that included both prep and total time. These were taken from the list and placed in a dictionary with the keys ‘prep’ and ‘total’.

For nutrition info, the ‘nutrient-name’ class actually included both the name and amount of the nutrient as one string, divided by a “:”. The name and amount were extracted by using the split method on the string (at the “:”). The nutrients were then placed in a dictionary with the nutrient name as the key and the amount as the value.

Load

The last step was to transfer our final output into a database, and it was determined from the beginning that a non-relational database would be the best option due to the nature of our data. At the start of our notebook, we established a local connection to MongoDB to create the initial database “recipe_db”. All the extracted data for each recipe was then placed in a dictionary, which was added to a collection named after the recipe category in our recipe database.

Summary

We took this approach so we could gather healthy recipes for people who want to make the transition to healthy eating, supplying people with new recipes for the respective category they want to aim for. The collections in MongoDB can be filtered to align with certain conditions people may want, such as:

- Meals for a certain category that is highest in protein (healthy eating and muscle gaining)
- Best healthy meals with the lowest cook times
- Highest ratings and reviews for the meals listed on the site
- An array of top recipes that supply the most amount of nutrients needed for daily consumption.
- Least amount of ingredients needed for the dish.
- Filter certain ingredients to avoid allergic reactions

These indices can be used to assist people who want to eat healthier with certain goals or requirements in mind.

We hope that the data we've collected is enough to sustain a start up company for the first year. We want to feature the highest rated and reviewed content that is found in the web and bring it into peoples homes at no cost of time.