Data engineering Project report

Abstract

Our project aims to recommend healthy recipes according to their benefits and effects on different parts of the body. We also wanted to add ingredients availability but we reconsidered the choice since there exists especial markets for finding ingredients for foreigner(ex: african groceries at "La guillotière").

Project steps

Ingestion

We retrieved a dataset from a previous study of recipes carried out by a team at INSA: Hummus data. This dataset contains details of recipes scrapped from food.com, along with author data and review ratings. We've added a few recipes from the spooncular api. The ingestion step is fairly simple. It consists of :

- A curl request to retrieve Hummus data from gitlab and store it in a csv file.
- The retrieval of 300 recipes from spooncular with diets and a filter on the request to get only healthy recipes.

Wrangling

Hummus data came with authors information and review we didn't want to use. So we removed columns related to them. We had also to sample because of the large amount of data, about 590 000 rows. Sampling was done among recipes having 'A' or 'B' as nutriscore value. Then we had to parse the csv file again because of some lines not well organized. In fact there were, for example, one long column value represented in several lines. The pandas read csv function split it into multiple lines causing data loss. The approach we did is reading each first string of a line and checking if it's an "Id". The lines which doesn't fit that rule were considered part of the previous line. Then we were able to use dropna() and drop_duplicates() without risking to have "false positive" lines.

After that we had to add diets information because hummus dataset didn't have this information. We first tried to fill that column by searching a match on the spooncular api. We did it by providing ingredients list and/or recipe's name and description. Unfortunately no matching were found. Then we tried to use Sparql requests to find the diet on Dbpedia. That didn't work too. Finally, diets are affected to a recipe using keywords in ingredients list. We used a regex search to do the matching. We also used exclusion lists to not, for example, class a recipe as vegan if its parsed ingredients list contains vegan keywords but also 'egg' which is not vegan.

Another part of our project was to link recipes with some body objectives. To do so, we tried to categorize them usings amount of nutrients within recipes. We used some recommendations of macro-nutrient proportions per day and per meal. The calculation approach is generalized and doesn't include detailed factors such as age, height, weight, activity level, and overall health. Here are values we used:

- Calories : Moderately active average a day for men 2600 and women 2000. A meal is 25% of recommended daily intake
- Fats : less than or equal to 30% of meal calories
- Saturated Fats : less than 10% of meal calories
- Carbohydrates : 55% of meal calories
- Fiber : at least 8g per meal, based on 30g per day
- Proteins : 15-20% of meal calories
- Sodium : less than 600mg per meal, based on 2400mg per day

Health objectives we have taken into account are : "Weight control", "Flat belly", "Blood pressure", "Nutritional balance", "Cardiovascular health", "Saturated fat reduction", "Carbohydrate balance", "Digestive health", "Diabetes prevention", "Balanced protein intake", "Muscle strengthening", "Blood pressure control"

At the end of wrangling, we stored our data on mongoDB collection.

Production

In this part, we chose neo4j to persist our data. Our choice was motivated by:

- There was no especial aggregation need : we din't have temporal data or need to aggregate values like sum up or calculate means to get insights.
- Our project goal is more a relationship purpose. Using graph database will help us perform complex queries without doing costly joins
- A future modification of the database will be simpler than in a structured postgresql in which that action could imply modifying schemas.

Improvements

Here are a few ideas for improvement

- The approach to determine diet using keywords is something we could do better. For Hummus dataset, since we knew in which website they came from and we noticed that recipes there were classed using diet, we could do further scrapping to complete hummus data with diet column.

- Also, a collaboration with a nutrition laboratory can help obtain a more exhaustive list of health Effects. And a work of training a machine learning model can be considered to determine the recipe's effects on health and the body.

- In the same way, with more data, a model could go further by proposing recipes that it has composed itself.