Team Leaflet

5/8/2021

Project 2



Fun with Foods!

A Brand Name Food Analysis of Nutritional Content using USDA Data

**Contents**

Theme……………………………………………………………………………………………2

Source and Transformations………………………………………………………………………2

Coding Approach………………………………………………………………………………….5

**Table of Figures**

Figure 1: get\_data() function for API call and processing. 3

Figure 2: Data object returned from USDA API call. 4

Figure 3: Code snippet foodNutrients.js (1 of 2) 5

Figure 4: Code snippet table.html 6

Figure 5: Foods table with search field for user input 7

Figure 6: Bar chart displaying food item's nutrients listed in grams. 8

Figure 7: Pie chart showing Vitamins and Minerals content of food item 9

**Theme**

We are studying nutrition and eating trends in the United States with an interest in improving public health. Our mission is to de-mystify nutrition science & make it more accessible by building an app that leverages publicly available, scientifically reliable, aggregated databases of nutritional data and eating trends to to empower Americans to make data-driven, conscious eating choices for themselves-regardless of their food preferences, allergies, health concerns, diet restrictions, or diet goals. Our app delivers up-to-date, highly detailed nutrition information alongside relevant, compelling visualizations that take the guesswork & mundane math out of nutrition science by crunching the data for any food input by the user, and delivering usable, health relevant insights immediately in a intuitive, compelling visualizations

Our main app is a web-page that allows the user to input their food of interest into a search bar on our home page. The user input will initialize our app to query, update, store & retrieve data on that food in our MongoDB database, while checking the data in our collection against the most recent entry on that food as listed in the USDA Central Foods Database, accessed by API call. Our app also generates colorful Plotly summary visualizations for the 2 most health-relevant, categorical nutrition profiles of any food: 1) the vitamins/minerals content (Plotly pie chart) and 2) the macro-nutrient breakdown (Plotly bar chart, plot by commonly diet relevant categories such as carbohydrates, protein, fat, fiber, fatty acids, etc.).

Our app returns data in both highly detailed table/textual form and visualized summation graphical form for maximum flexibility, broad utility & personalized function, so that it can serve a diverse range of nutritional needs. Nutritionists, dietitians, people with gastro-intestinal conditions,, and those with serious food allergies may be looking for specific information found in a detailed ingredients list.. People trying to lose weight or looking to customize a diet for a specific purpose, such as athletes, keto/paleo dieters, and vegans, will love the convenience of the macronutrient breakdown. Pregnant women, humanitarian food aid organizations, and medical professionals or patients looking for foods to help with any one of a number of conditions that cause or result from a critical mineral/vitamin deficiency can glean the information they need from 1 look at the pie chart.

Beyond the nutrition app, we wanted to study current eating trends in the United States and see if we could find mappable correlations by geography

For food trends, we chose to visualize Farmers Market locations nationwide and evaluate their properties with socioeconomic data such as fast food restaurant growth on a map. We hoped to find geographic, mappable correlations between ecological, economic and environmental variables across the country.

**Sources**

**USDA Food Data Central (API)**

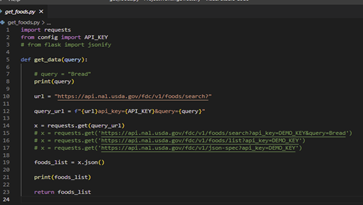
This source allowed us to assess nutritional and food data. It was designed for people like us, those working with data, to display food and nutritional information on websites and apps.

**USDA National Farmers Market Directory (.csv)**

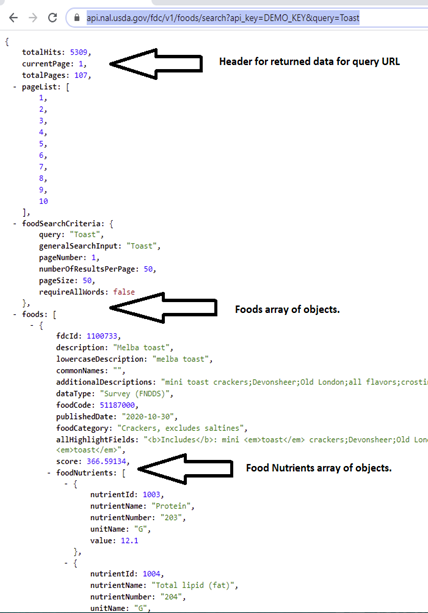
This source provided a list of farmers markets in the United States, at which agricultural products are sold on a regular basis.

**Transformation**

Data from the API call to the USDA foods database was returned as a data object. This object was converted into JSON format for input into and later retrieval from, the MongoDB instance “foods\_app”.



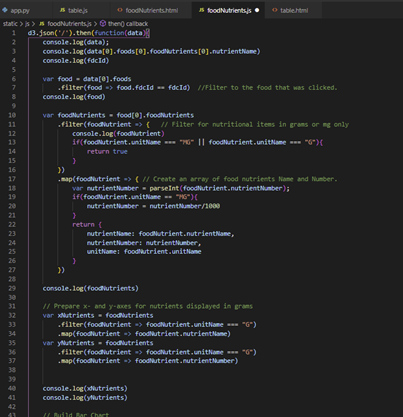
*Figure 1: get\_data() function for API call and processing*

**

*Figure 2: Data object returned from USDA API call.*

The JSON object was then made available to the JavaScript files “table.js” and “foodNutrients.js” via D3. Using JavaScript in the “table.js” file, each item of food in the returned data object was called and its brand name and ingredients list written to a table in “table.html”.

In “foodNutrients.js” the same data object was processed to retrieve the array of objects contained within each individual food item by passing the parameter “fdcId” in the “/foodNutrients” route of the “app.py”. This returned the first food item that met the ID criteria (it being an FDC identifier). Once the food item was retrieved the object array foodNutrients was called and filtered for those entries that were displayed in mg or grams. Further nutrients (i.e. Energy in kjoul) contained in the array were excluded from the following visualizations for consistency purposes. Of interest to the consumer is certainly the amount of nutrients one can expect when purchasing a certain brand of food item when compared to another.

****

*Figure 3: Code snippet foodNutrients.js (1 of 2)*

When clicking on a particular food item in the foods table, the user is forwarded to the “foodNutrients/<fdcId>.html” page, where the file “foodNutrients.js” provides the data handling necessary and the preparation code (via Plotly) for the visualizations displayed.

**Coding Approach**

**The main integrated body of our project is a web-page with a home page with multiple routes rendered by the flask app in our Python.app script. The web-page responds to user input requesting the complete information on a particular food item (includes brand, ingredients, and nutrition information) and performs an API call to the USDA foods data central API database in order to retrieve the most up-to-date information, utilizing the MongoDB database acting as an intermediary storage depot where the app stores, updates, & retrieves data in collections.**

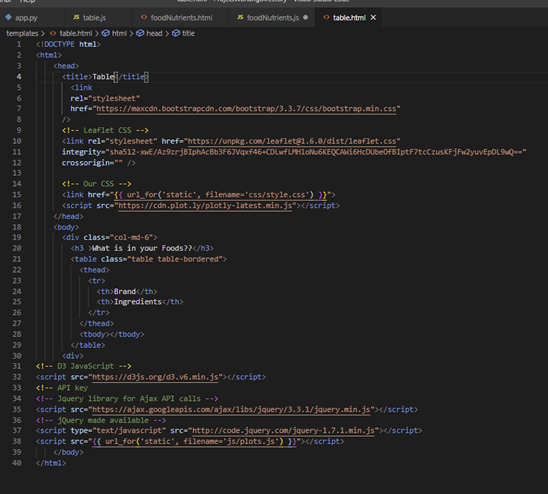
Our page layout is written using Javascript with html & css styling, with the home page & routes organized by our flask app. **We use the flask jsonify function to perform API call get requests initiated by user input requesting information on a particular food item, and to mediate the packaging & delivery of the response data as it travels from the USDA foods database to our MongoDB database collections before being displayed on the front-end webpage as user output for the food item request initiated in the UI.** A food variable in our script holds the information from the user-submitted food item description so that our app can search the USDA foods database for matching results. The returned food data includes brand, ingredients, and nutrition information published into a table.

Our app contains built-in functional flexibility for how it handles, stores, and displays quantitative entries in the nutritional information. The script controls how the app handles units for qualitative nutritional data via the foodNutrients variable, and so the developer can easily change the units with a simple modification to the variable at any time. The app has been tested & verified to work using both grams & milligrams-we chose to use grams as our chosen unit throughout the app & in our charts & graphs, where we plot aggregated ingredient/nutritional data to visualize broader trends & categorical insights revealed by our analysis of all the data on individual food items found in the USDA database. These visualizations are integrated into our web-page app interface on the front end by additional flask routes in our **python.app** script. The app stores the food nutritional data in arrays, referenced by both a unique Name (string) and Number (integer) as identifiers for each entry that can be returned & queried as a search result.

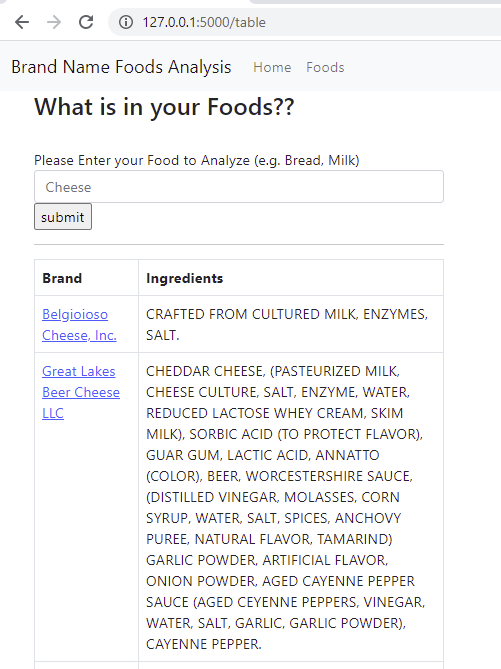
The app stores, updates, and retrieves data both to and from our MongoDB database, where it stores the latest data on each food item in a collection until the user updates the information by initiating an API call with a front-end query on the web-page UI.

Finally, we used D3, JS, Leaflet, and Plotly to display our visualizations. We built both a bar chart and a pie chart using Plotly, visualizing broader trends and summative quantifications of selected nutrients of interest across all aggregated, queryable food items in our app database.

We named the page “Fun with Foods!” to describe our project. We used bootstrap to customize the layout and style of the page. In our index.html, we also created a collapsible navigation bar and search option. The user can enter a specific type of food (for example, bread) and hit submit. Our app will then display the nutritional content of the food entered.

****

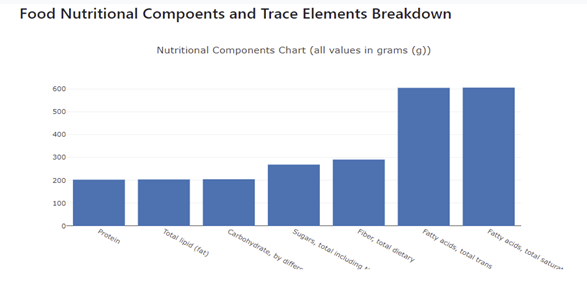
*Figure 4: Code snippet table.html*

****

*Figure 5: Foods table with search field for user input*

**Visualization 2- Nutrients Bar Chart**

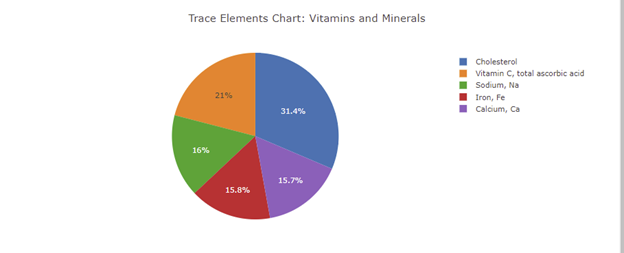
The bar chart shown below utilizes the USDA FDC API to visualize different types of nutrients in grams. A new bar chart is generated for each food that is selected on our tool.

****

*Figure 6: Bar chart displaying food item's nutrients listed in grams*

**Visualization 3- Vitamins Pie Chart**

The pie chart below is an example of the pie charts that can be generated for a food selected on our tool. Our tool creates a pie chart of 8 vitamins (cholesterol, niacin, riboflavin, thiamin, sodium, potassium, iron, and calcium) that are present in the chosen food.

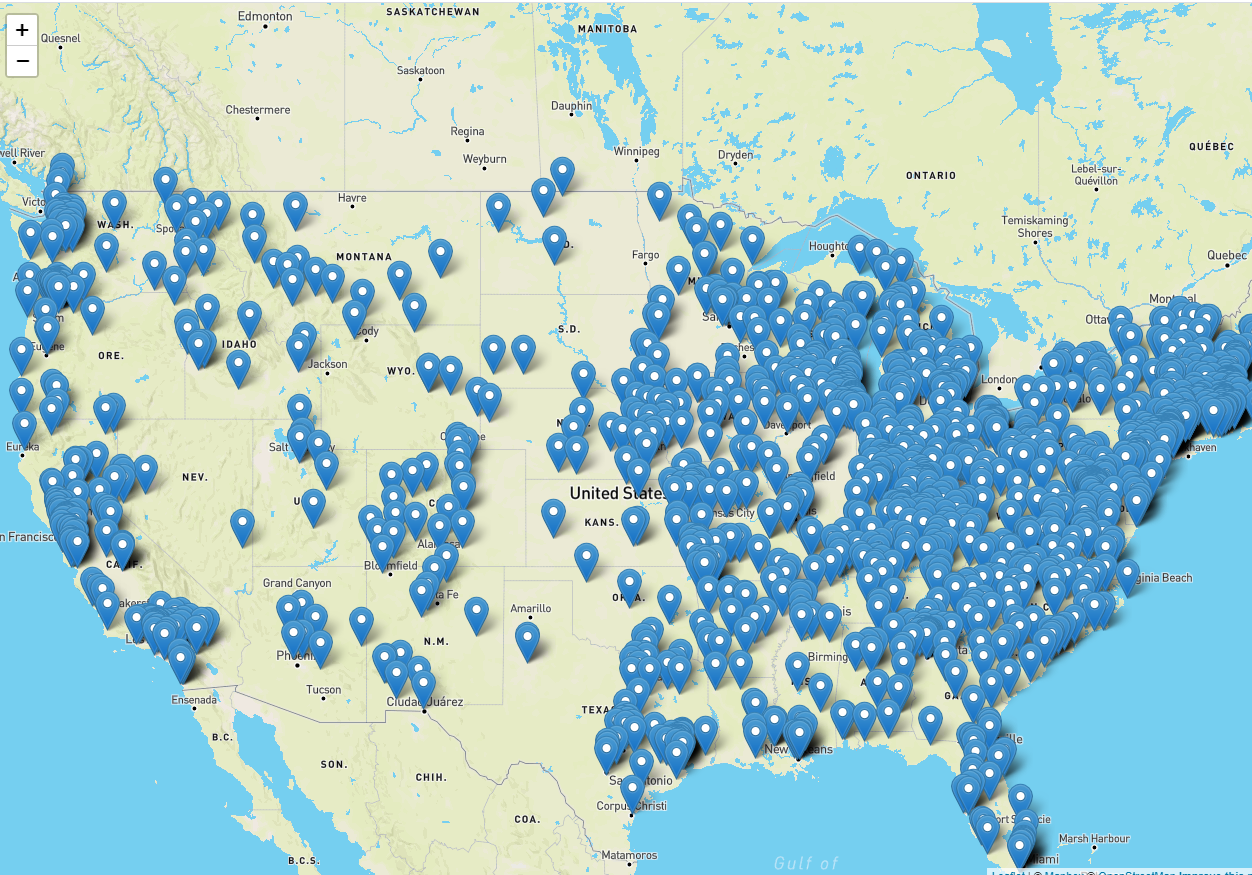
****

*Figure 7: Pie chart showing Vitamins and Minerals content of food item*

**Visualization 4- USDA National Farmer’s Market Directory**

Next, we wanted to visualize American diet trends by geography across the country. We started by obtaining a list of over 3,000 farmers’ markets in the United States from the National Farmers Market Directory via the USDA Agricultural Marketing Service. The data included detailed information on each market, including name, location, and food availability across broad and specialized categories.

We wrote Python scripts utilizing the Pandas library to read, clean, and organize the farmers’ market data for mapping in Leaflet, eliminating locations without coordinates and paring down the properties of interest into just a few broad categories of food availability, such as whether the marker offered organize items, fruits, vegetables, meats, and whether they accepted WIC/SNAP funds, which are government food assistance benefits. After cleaning the data, we also used Python to complete the farmers’ market data transformation from csv into cleaned Pandas dataframe into an output geojson file, type “feature collection.” We were able to turn our geojson file into a Leaflet-compatible javascript file by setting the geojson object equal to a variable and calling the variable in leaflet code to plot the entire geojson object onto the map.

****