Homework 2 (HBase) - Part 2

Dave Millman
Ben Polk
CSCI 540 - Fall 2021

October 21, 2021

The question I sought to answer for this part of the assignment was, "What types of food might be available at grocery stores in the Mississippi Delta region, and what is their nutritional content?" I came up with this question after finding the "Delta Food Outlets Study" dataset on the Data.gov website.[1] This dataset contains the results of an observational tool used to catalog various aspects of grocery stores, convenience stores, and restaurants around the Mississippi Delta, including what foods are available at the location.

This question seemed interesting for two reasons. Firstly, I wanted to see if it was possible to predict which foods from the MyPyramid dataset might be available at each grocery store. Determining what items are or aren't available at a store is a common question with many practical applications. Second, I wanted to be able to view a nutritional profile of a grocery store, based on its possible available foods. This is interesting because such questions can aid research into geographical areas that may be nutritionally underserved.

However, using the Delta Food Outlets Study data to help answer this question required some preprocessing. First, I had to examine the 'Data-Dictionary' file to determine which questions in the survey corresponded to the availability of what foods. I then translated the food descriptions from this study into food 'keywords', that could be used to perform lookups on the MyPyramid food data. (This translation can be viewed in the file

[1] This dataset can be found and freely obtained at https://catalog.data.gov/dataset/delta-food-outlets-study; it was compiled by the Department of Agriculture Agricultural Research Service.

"mapped_availability.xlsx".) Finally, I built a small processor in Java that reads in the raw survey results in CSV format, translates the question answers into food keywords, and then outputs the results in XML format. (The processed file is "store_food_data.xml").

My HBase schema was very similar to the one I used for part 1. A combination of food name and portion was used as the key, and nutritional data was added to columns in a 'nutrition' column family. However, this table included an additional column family, 'stores', to facilitate tracking which stores might have the food item described in the row key. To populate the table, I first read the MyPyramid food data into the table. I then read the store food data. In the XML file, each store has a list of food keywords as indicated on the survey; the presence of a keyword means the food was available at that store. For each food keyword, I perform a scan on the table to find any row keys that contain the keyword. If a row key does contain the keyword, I add the store's ID as a column to that rows 'stores' column family, with a value of 1.

The resulting, final schema has food and portion names as the key, a column family for nutrition values, and a column family for stores. The 'stores' column family has one column per store that might contain the item (based on a match between a keyword and the row key). I liked this approach because it allowed me to flexibly track which stores have which items; different rows can have different columns representing store availability. It is also easy to scan the table to find rows that have certain columns, in order to find all the foods that might be available at a given store.

Obviously, this approach is somewhat naive in terms of predicting the presence of a food item from the MyPyramid dataset based on a match with a keyword assigned to the Delta Food Outlets (DFO) dataset. For example, if a store in the DFO dataset indicated 'corn' was available, then I attempted to find any row keys that contained that keyword. This would return rows such as 'corndog', which is obviously quite different than an ear of corn. Still, the store would have shown as having corndogs available. Some accuracy was sacrificed in order to get a working proof-of-concept.

To use this system, the user would first need to run the supplied 'create.rb' and 'load.rb' Ruby scripts for the HBase Shell, which will create the table and load data into it. Then, there is a file named 'query.rb', which demonstrates the queries I used to answer the question for grocery store 'g001'. In

order to find information about other grocery stores, it would be necessary to modify 'query.rb' to use a different store id (which range from 'g001' to 'g012').