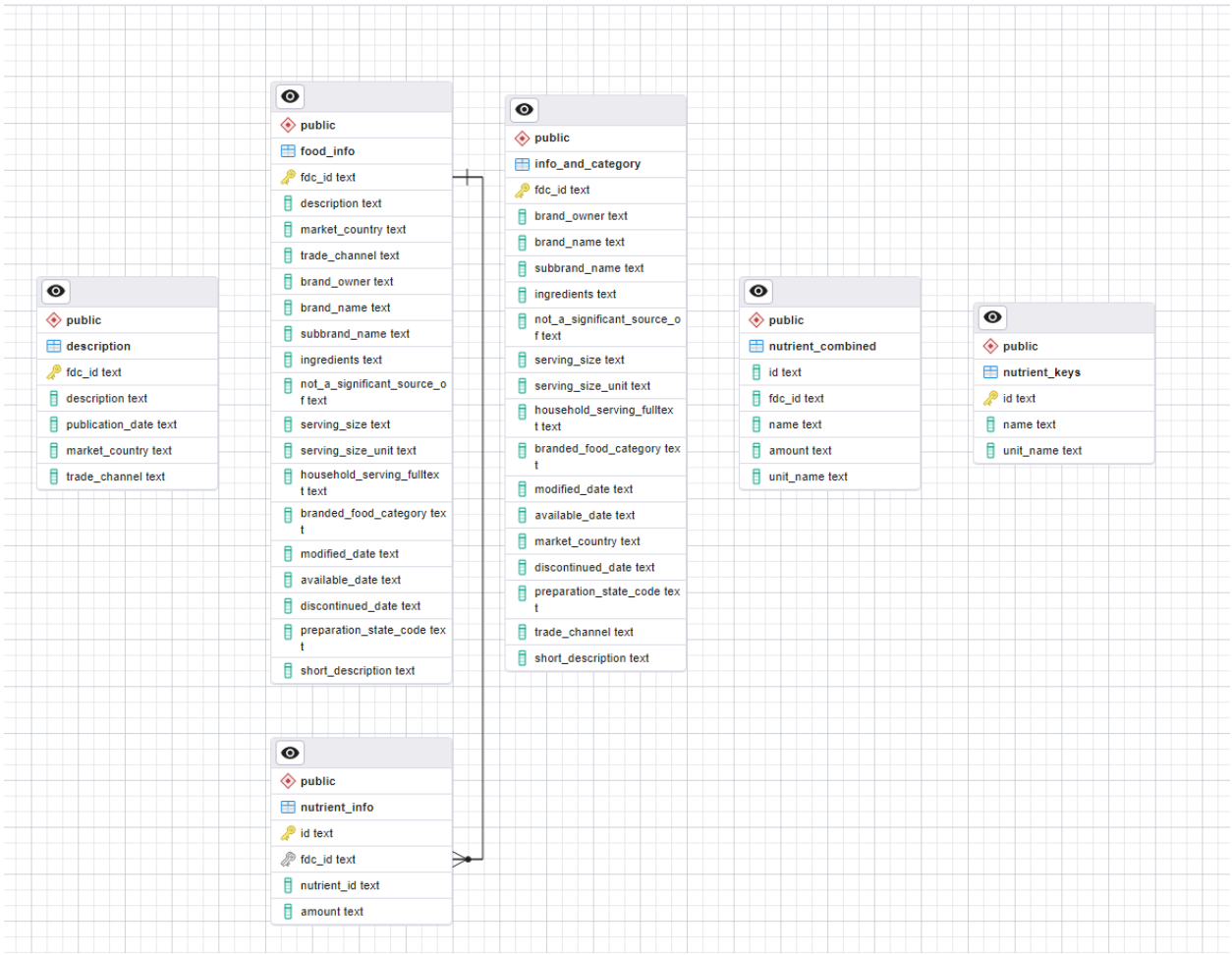
USDA Branded Food Tier List

Summary

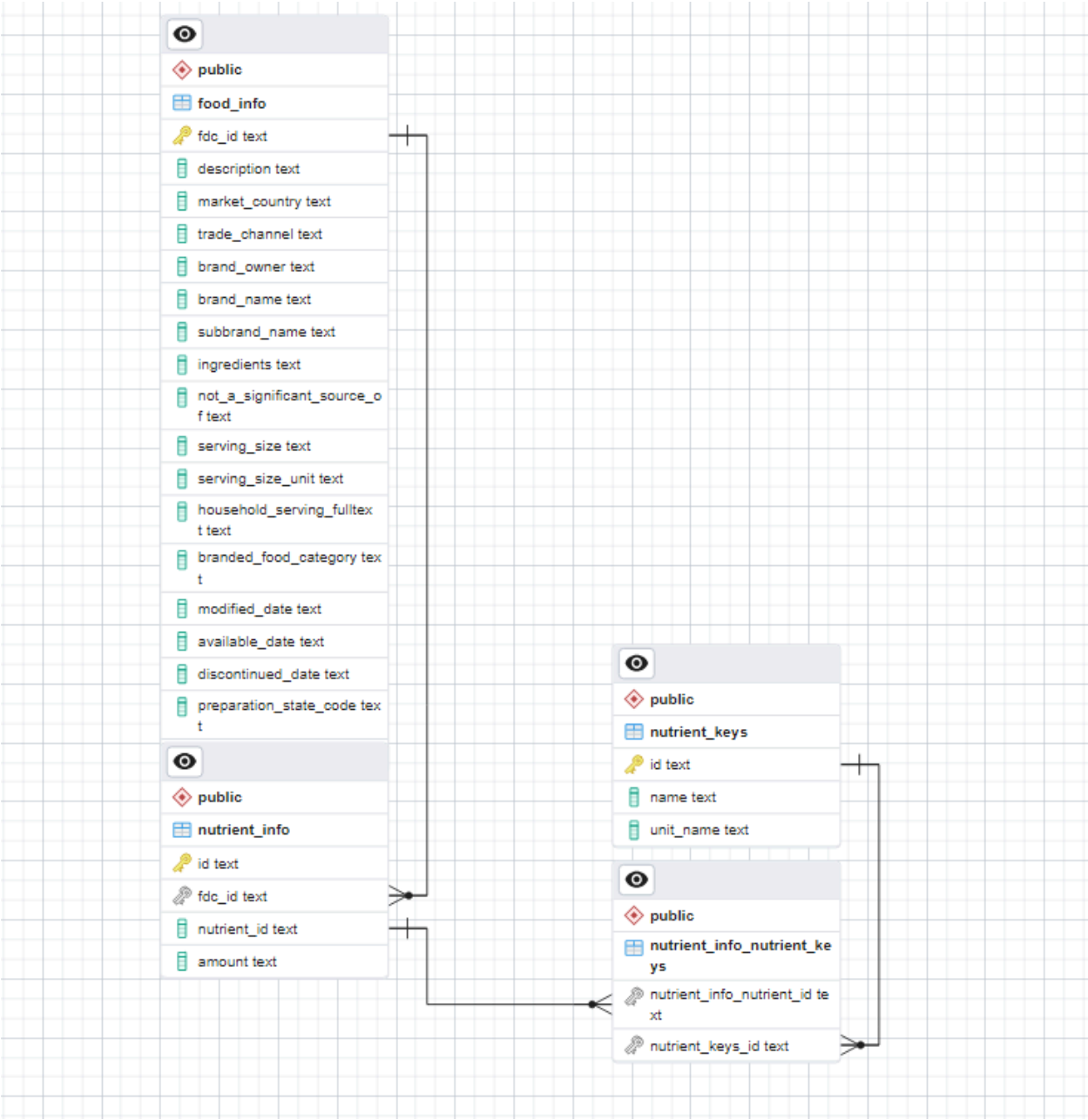
Using USDA (United States Department of Agriculture) data on branded foods, I wanted to create some sort of basic ranking/scoring system by food type and overall. The scoring method was based on percent daily values for various nutrients such as fat, sugar, and various vitamins. I used a simple but flawed equation consisting of the proportion of the percent daily values for each of the nutrients and averaging them out (more on the scoring later). Using Kernel Density Estimate graphs (to see the distribution), I learned that most of the food categories' scores had some sort of bimodal distribution. I suspect there is some correlation with the prices of the foods and the existence of some price threshold for each of the food categories. However, a more solid scoring equation(s) would be needed to make sure that there isn't any spurious correlation. If another scoring equation(s) generates similar types of distributions for the various food categories, then one could proceed to research the relationship between the scores and the prices with less worry.

Data Structure (SQL → Jupyter Notebook)

From the USDA website, the branded food data contained multiple excel files.



Here is the following entity-relationship diagram showing the relevant tables before preprocessing:



Here is the entity relationship diagram after cleaning the data

Food_info has a many-to-many relationship with nutrient_info and nutrient_info has a many-to-many relationship with nutrient_keys. The reason being is because multiple rows can be associated with multiple nutrients.

Setting up the scoring method

I used the following nutrients: protein, Vitamin A, Vitamin D, Vitamin C, calcium, magnesium, iron, potassium, total fiber, saturated fat, zinc, sodium, total fat, cholesterol, trans fat, added sugars. I found the percent daily values for each of the nutrients and used the following equations depending on if they are considered “good” or “bad” for you.

Good → 1 - ((recommended - nutrient_value) / recommended); if $> 1 \rightarrow 1$

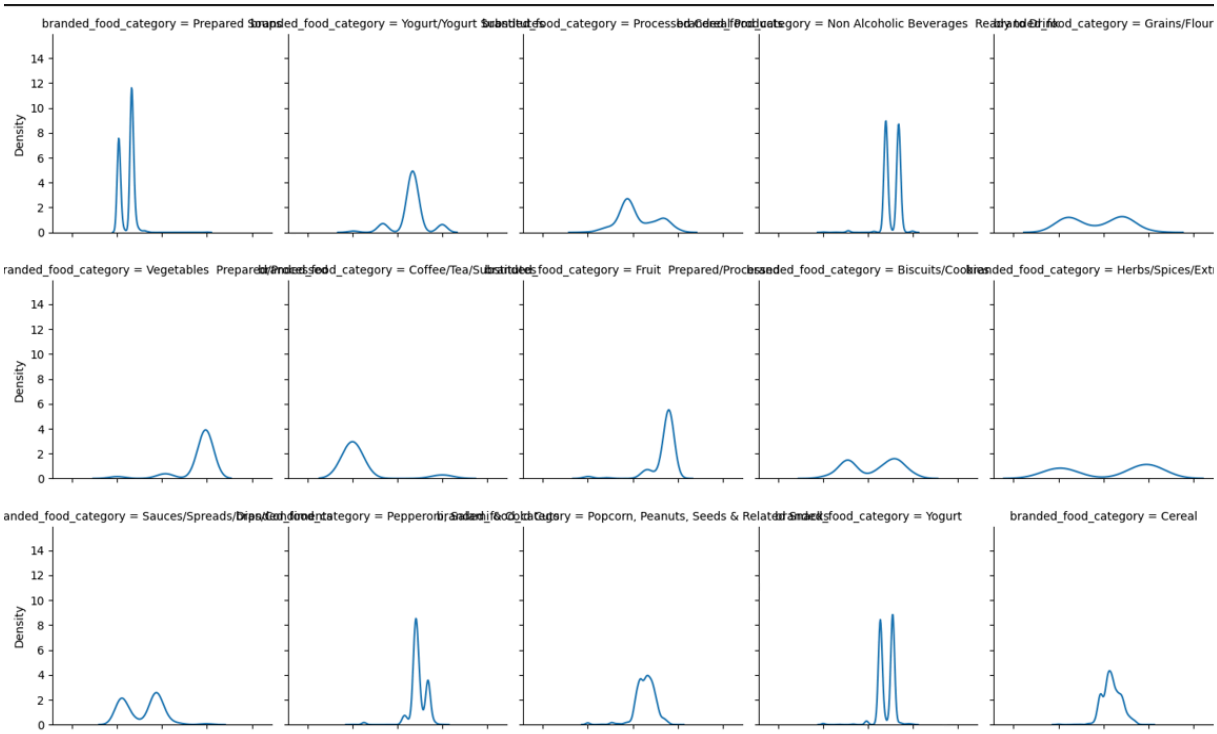
Bad → ((recommended - nutrient_value) / recommended); if $< 1 \rightarrow 1$

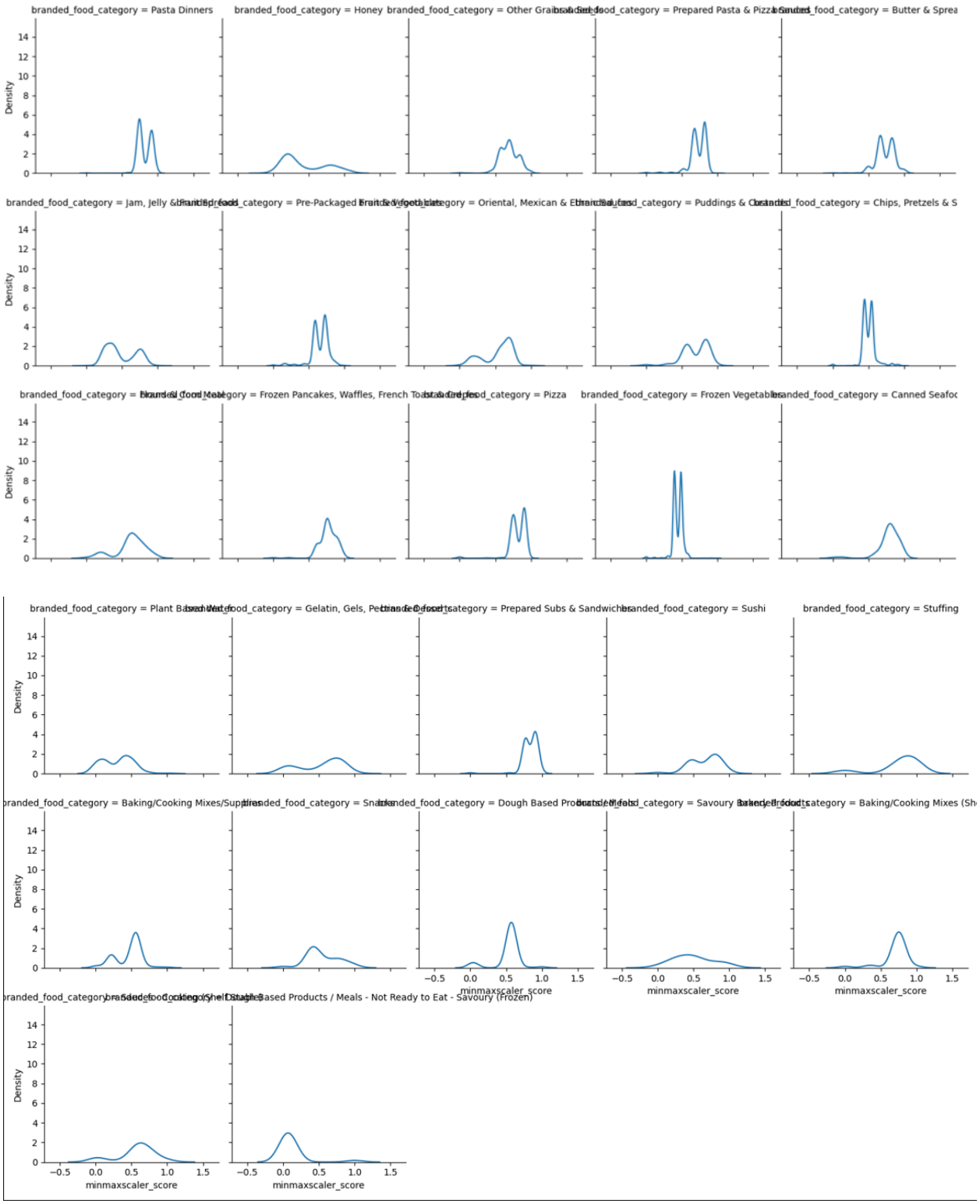
Reasoning: For the good nutrients, you want the actual nutrient value to be close to the recommended. For the bad nutrients, you want the value to be close to 0. However, a major flaw is that for nutrients with greater recommended values, a nutrient with a lesser recommended value would be weighted less given equal differences (recommended-nutrient_value). Also, Vitamin A and Vitamin D had multiple columns due to unit differences. To resolve this, I divided the total amount of nutrients used and subtracted by 2 as a naive solution.

For the tier list I implemented, it’s a straight up grading system. [0.9-1] scores would be considered an A. In order to use this type of tier list, I used MinmaxScaler from the scikit-learn package. MinmaxScaler basically just shrinks the scale of the data so the lowest score would be 0 and the highest score would be 1.

Distribution of the Scores

Here are some graphs of the scores for different food types (names are too long)





Lots of the graphs have some sort of hilly/bimodal shaped distributions. It would be interesting to compare these graphs with the prices of the foods.

Tier List

The tier list can be best viewed on the Streamlit app I created. You will be able to sort and filter various columns.