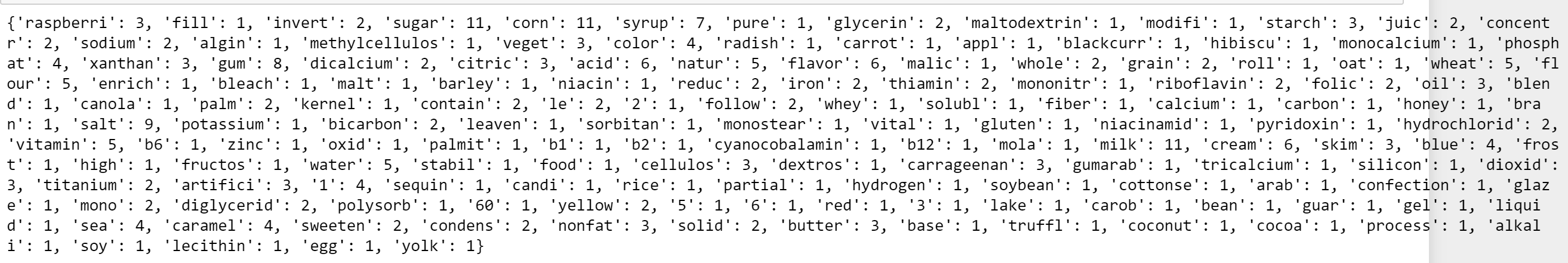
Christopher Singh Data Science Project

1. Identify the unique ingredients.

The way to identify the unique ingredients in Python would be to traverse the entire datasets and use key-value pairs to keep track of the counts of ingredients. This will output a dictionary in which the ingredient is the key and the count is the value.



This snapshot was taken from the jupyter notebook file. There is still some data cleaning that needs to be

done because not all of the dictionary keys are ingredients. For example, the keys named color and follow

are not ingredients but they still appear in the dataset.

I obtained this list of unique ingredients by using the NLTK package to clean the data. I removed stop words,

I applied stemming and lemmatization as well as collocations.

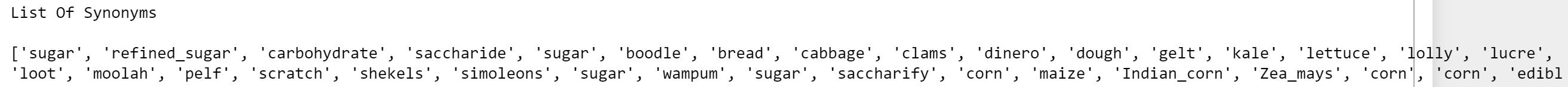
The unique ingredients would be: **raspberry, sugar, corn, syrup, glycerin, corn starch, juice concentrate,**

**vegetable juice, xanthan gum, whole grain oats and flour, vegetable oil, corn fiber, carbonate honey, salt,**

**molasses, milk, cream, skim milk, fructose corn syrup, water, sequin candy, cocoa, coconut oil, sweet condensed milk, butter, caramel, eggs, cellulose gum, rice flour, carob bean gum, sea salt caramel, liquid sugar, soybeans**

2. Identify synonyms.

The way to identify the synonyms in Python would be to use the Wordnet module of the NLTK package. The wordnet package can be used to generate synonyms as well as antonyms. The way that I used it in the code was that I passed the clean data list as a parameter and stored the output in a dictionary.



This small snippet shows that synonyms that were generated for sugar, corn, bread, etc. You can view the entire list that was generated in the Jupyter Notebook file.

Synonyms: **sugar/glucose/glycerin/confectioner, salt/sodium, sweet/sugar/syrup, water/liquid, filling/puree, zinc/iron, wheat/gluten, corn starch/corn flour, honey/molasses, fructose/cellulose, thiamin/vitamin, xanthan gum/guar gum, sea/water, soy/whole wheat, nonfat/skim, iron/fiber, cream/butter,**

3. Identify descriptors.

My understanding of descriptors from a linguistics point of view, is that they provide additional information

to a topic or noun.

Possible descriptors: **modified, natural and artificial flavor, color, soluble, sweet, processed, reduced, blue frosting, raspberry filling, partially, contains less than, vital, vegetable juice for color, titanium dioxide for color, nonfat milk solids**

In terms of Python, descriptors tend to implement both set and get methods of object oriented programming.

If we were implementing the object-oriented paradigm, I would choose the following for an Ingredient class:

Set and get methods for Ingredient\_Name()

Set and get methods for Nutritional\_Value()

4. Approach for a list of 300,000 ingredients.

My approach for a list of 300,000 ingredients would be to create a Jupyter Notebook similar to the one that is attached and rely on Python dictionary key-value pairs to identify the key ingredients. I would set a counter to identify the most commonly used ingredients as well. Additionally, I would create a corpus to set linguistic features to the list. I feel that this is necessary mainly because the list is not entirely ingredients. There are descriptors that need to be taken into account as well. The ingredient would be a noun while the descriptor would be an adjective.

The tools that I would use for this would be the NLTK package and Spacy. Both packages provide great NLP power which allows me to analyze the actual text and extract meaningful insight from it. I would then build out an LDA Topic Modeling machine to identify the most important topics and to see how each topic correlates to other topics. The LDA model that I built out used only a small amount of data but I feel that if I had a larger dataset, I would get a better overall understanding of the various topics.

The next tool that I would implement would be the sentiment based analyzer. This would allow me to determine whether an ingredient has positive or negative connotations. This would inform the user that an ingredient may or may not be good for them. For example, sugar and salt would be negative if the user has underlying medical conditions such as diabetes or high blood pressure. This type of insight can influence the purchasing habits of individuals when they are shopping. The sentiment analyzer that I implemented was able to describe the cleaned version of the sample dataset using Naïve Bayes classifiers.