Jack Morgan

5 December, 2021

ID - 00001360442

CPID Product Similarity and Frequent Chemical Combinations

The household products that millions of Americans use daily are typically a mixture of many different chemicals, each with their own unique health and safety hazards. Products can be irritants, dangerous when inhaled, highly flammable or reactive when combined with other chemical solutions. To help protect and inform consumers, the Consumer Product Information Database attempts to present product information in a human-readable format, allowing consumers to make informed decisions about what products to purchase and use.

For this project, I have scraped basic product information from the Consumer Product Information Database in order to find frequent sets of product ingredients, find what form (liquid, aerosol, solid, etc.) products come in as well as build a tool to measure the similarities between multiple product ingredient lists.

My implementation of these tasks, for the most part, use standard Python libraries, however there are three external libraries that are required—requests for making HTML requests, BeautifulSoup for HTML parsing and apyori for a simple Apriori algorithm implementation. A code sample to install all three of these packages is included at the top of the submitted Ipython Notebook files. Additionally, fuzzywuzzy was used as a basic string matching tool to ensure that there were no duplicates or close matches in certain product attributes, but it is not used in any of the analyses.

The first step I took to scrape information from the Consumer Product Information Database was to compile a comprehensive list of URLs linking to each individual product page.

To accomplish this, I used the website's search function to list products beginning with each letter of the alphabet (in addition to a '0-9' search filter for numeric products). On each page, product links were held in an organized table, however some cleaning was necessary to collect proper links, as the ones referenced in the href values contained whitespace, line breaks and other HTML tags. To remove whitespace, the link was separated using Python's split() function, creating a list of characters. This list was joined back together on an empty string, resulting in a link with no whitespace or line breaks. To clean the HTML tags from the links, a regex function was used to locate angle brackets for removal. For the purposes of this project, products which are old or off the market were not considered, so any grey shaded links (indicating an old product) were not collected. In total, about 14,000 links were collected in the 'product_links.txt' file.

With the product links collected, it was then possible to write a program to scrape information from each page. For the sake of simplicity, the fields collected for this basic analysis are—Product Name, Product Categories, Usage, Form, Manufacturer, Chemical Ingredients and URL. With more time and a larger scope, it would be interesting to look at hazard information, first aid, health effects and chronic health effects, however this data is not present on all pages. After collecting all of those fields (where available), the scraper would move on to the next product link. To scrape the 14,000 URLs was a fairly long process, taking about 6 hours. This information was then written to 'prod_details.csv'.

After collecting some basic information on each product listed on the site, it was then possible to perform basic analysis on the products, build scripts to run the Apriori algorithm on product ingredients, as well as find cosine similarity values between any set of product ingredients.

For the purposes of this project, all information on chemicals found in products is stored in the form of CAS numbers. These identifiers are issued by the American Chemical Society, creating a registry of chemical substance information. Each is a 10 digit number which provides a unique identifier for any ingredient present within the product set. When performing these sorts of data tasks, it is usually more efficient to work with unique ID numbers rather than raw strings, as string matching is both less computationally efficient and can cause issues when matching data points together. While scraping the Consumer Product Information Database, I initially gathered raw strings for chemical names, but found rather quickly that missing information, malformed strings and capitalization mistakes could cause difficult-to-debug errors.

When looking at the product information available, a few of the questions which spring to mind include—

“What companies are making the most products?”

“What kind of products are these companies making?”

“Which companies are creating products with harmful ingredients?”

The companies making the most products are typically the larger companies making general household goods.

Company	Amount of Registered Products
Procter & Gamble Co.	1461
SC Johnson, Inc	834
Reckitt, Inc.	547
Unilever	484
The Clorox Company	298
Sherwin-Williams Company	279
The Dial Corporation	247

The Glidden Co.	226
Rust-Oleum Corp.	213
Church & Dwight Co., Inc.	208

Top 10 companies with the most registered products

As each of these registered products is also associated with a list of categories for easy product classification, it is also possible to find the top categories for each of these companies. For the sake of simplicity, I will only include the top 5 companies and the top 3 categories per company. An expanded version can be found by running the code in the included Notebook.

Company	Category	Amount of Products in Category
Procter & Gamble Co.	Personal Care	975
	Inside the Home	439
	Hair Care	345
SC Johnson, Inc.	Inside the Home	702
	Air Freshener	253
	Cleaner	222
Reckitt, Inc.	Inside the Home	423
	Air Freshener	180
	Cleaner	147
Unilever	Personal Care	481
	Personal Cleanliness	205
	Hair Care	180
The Clorox Company	Inside the Home	199
	Cleaner	162
	Hard Surface Cleaner	156

Top 3 product categories for the top 5 companies with the most registered products

As discussed earlier, an advantage to saving ingredients as CAS numbers is that it becomes easier to link together data from multiple sources using the same identification standard. Within the CDC, the National Institute for Occupational Safety and Health (NIOSH) maintains a master list of chemicals considered hazardous in a work environment. This data is displayed in a table online, with CAS numbers and the associated chemical name. To gather this data, I built another small scraper, simply to scrape and store CAS numbers and chemical names. This data is useful, as the CAS numbers for hazardous chemicals can be cross-referenced with the CAS numbers found in product ingredient lists, helping to determine which products could be considered hazardous if misused.

When calculating the proportion of products containing harmful ingredients per company, companies which had created fewer than 50 products were not considered, as smaller, more specialized companies will likely create many harmful products which are not intended for personal use. Somewhat unsurprisingly, the companies creating the highest proportion of potentially harmful products are typically manufacturers of paints and coatings. What was surprising to me was that the other category of companies creating high proportions of potentially harmful products were makeup brands.

Company	Proportion of Products Containing Harmful Ingredients
Krylon	100.00%
L'Oreal Paris	100.00%
Kelly-Moore Paint Company, Inc.	99.07%
Benjamin Moore & Co	97.44%
Rust-Oleum	95.77%
SprayWay Inc.	95.70%

Red Devil, Inc.	95.15%
Revlon Inc.	94.00%
The Glidden Co.	93.81%
The Valspar Corporation	91.84%

The 10 companies with 50+ products on the market and the highest proportion of products containing harmful chemicals

Company	Proportion of Products Containing Harmful Ingredients
Gamblin Artists Colors Company	12.24%
Natural Chemistry, Inc.	13.46%
Duncan Enterprises	13.49%
Scotts Company	21.94%
Epson America, Inc.	22.64%
Bonide Products, Inc.	23.16%
Voluntary Purchasing Groups, Inc.	30.51%
Sopus Products	35.56%
Elmers Products, Inc.	36.29%
Meguiars, Inc.	44.88%

The 10 companies with 50+ products on the market and the lowest proportion of products containing harmful chemicals

The Apriori algorithm implementation is fairly simple, using the apyori library to run the algorithm. In this scenario, our ‘list of lists’ is a list containing lists for each product’s ingredient set (represented by CAS numbers). Perhaps unsurprisingly, many of the frequent itemsets involve water and artificial fragrances, both of which are extremely common in household products. The following table contains some of the more interesting frequent item rules.

Rule (LHS)	Rule (RHS)	Support	Confidence	Lift	Product Categories
Propane	Butane (Hazardous)	6.06%	95.17%	11.45	Air Freshener, Rust Proofer
Methylchloroisothiazolinone	Methylisothiazolinone	6.39%	96.44%	9.86	Shampoo, Exfoliator
Sodium Chloride	Methylisothiazolinone	3.56%	47.21%	4.82	Laundry Detergent
Citric Acid	Glycerin (Hazardous)	3.63%	96.76%	2.18	Hand Wash, Color Shampoo
Titanium Dioxide	Calcium Carbonate (Hazardous)	2.50%	49.86%	3.47	Grout, Acrylic Paint, Caulk

5 Example itemsets (MinSup = 2.5%, MinConf = 40%)

In addition to implementing the Apriori algorithm, I have also implemented a system which determines the cosine similarity between two products. As the ingredients list can be represented as a vector of chemicals with a 1 representing the presence of a certain chemical, product similarity can be calculated much like document similarity. To make this system more useful in comparing a group of objects, links to CPID product pages can be added to ‘urls.txt’. These links are then put in combination pairs, each of which are inputs to a cosine similarity function. A good use case for this would be to compare ingredients for multiple similar products. For example, there are 10 registered products which are labelled as exfoliants. By placing these 10 product links in ‘urls.txt’, the product pairs can then be sorted in order of similarity (try running the cosine similarity code block!). Product ingredients are listed for each result and hazardous ingredients are labelled as such.

Note that URLs must be taken from the ‘prod_details.csv’ file

While this project is fairly simple, there is a lot of room to extend its functionalities and incorporate data from multiple sources. With time to widen the scope of this project, it could be interesting to pull in more specific information on chemical ingredients instead of simply marking them based on whether or not they are harmful. Products could be sorted on toxicity, flammability and health effects. Multiple sources of information could be considered when labelling ingredients as harmful or not. As an exercise in data scraping and basic analysis, it was enjoyable and interesting to see what goes into the household products that I use on a daily basis.