Group 1: Camille Postaer, Laxus Nikolaev, Joey Secard, Addie Hermstad
Winter 2022
California Polytechnic State University MSBA

World Bank PPP Project Report

The World Bank presented us with the large and important problem of optimizing the Purchasing Power Parity calculation process. The Purchasing Power Parity gives economists insights into a countries' standard of living and economic standings and is therefore a very valuable metric. Currently, the process can take upwards of four years to calculate, because pricing data is being collected manually by surveys distributed across the globe. The World Bank's ICP team expressed its difficulty in matching similar products within the data collection and comparison process.

The Purchasing Power Parity considers a large array of product categories from all around the world. Due to our time constraints, we knew that tackling multiple product categories would not be feasible. We also realized that actual data collection was not the most important problem for us to solve. We believed that a more valuable issue to tackle was the matching phase. Our group decided to focus on US over-the-counter pharmaceutical data because of the importance of pharmaceutical products in the Covid-19 era. The World Bank had also previously expressed their difficulty in matching pharmaceutical data. If our group was able to successfully match one of the hardest product categories, we hoped our processes would be easily replicated across other categories and a wider range of data.

The goal of our project was to create a multilayer, or "tree-like" classification model for classifying pharmaceuticals. The first layer of our tree would classify drugs down to one of five categories: cough/cold/flu, kids' medicine, allergy/sinus, pain/fever, and digestion/nausea. This layer of our tree would be determined using the description of the usage of the drug. The next layer of our tree would further classify our drugs to distinguish whether they are generic or brand names. To do this, we created a code that would classify each drug as brand-name if the brand's name was in the product's name and generic if not. Of course, whether or not the brand name was in the drug name was not an indication if the drug was generic or not 100% of the time, but we found it to be the most reliable indicator.

The first step in our project was sourcing our data. We chose to work with pharmaceutical data from Target.com because it was easily accessible and gave us access to over 400 pharmaceutical product data entry points. To scrape from Target.com, we used a free web scraping tool called ParseHub. ParseHub used loops to scrape through every page on the website that had pharmaceuticals on it. It then allowed us to code the link access to navigate to individual drug details. With this tool, we were able to scrape drug names, brands, use descriptions and prices. We scraped our data by drug category: cough/cold/flu, kids' medicine, allergy/sinus, pain/fever, and digestion/nausea. This left us with five datasets.

The most time-consuming portion of our project ended up being cleaning our scraped data. Once scraped, we had to compile all the data into one usable csv file using the rbind() feature in R. We had to make sure that all columns lined up and were labeled identically before this was possible. Once compiled, we removed any unnecessary information and null values.

Next, we began the task of identifying frequent and important words contained in the drugs' usage descriptions. We wrote an algorithm that gave us the top 10 words (ignoring stop words) contained in the drug usage descriptions in each category. We found a total of 35 words to use in

our model as dummy variables.

This graph shows the most frequent 10 words in our dataset and what percentage of the word's occurrences fall within each category. See the appendix for this graph expanded to all 35 words used as dummy variables.

Once we had our data cleaned and indicator words identified we were able to run models in R. We first ran our data through an unsupervised data analysis technique called Principal Components Analysis using the princomp() function. This analysis gave different weights to different words, thus ranking their importance in classifying the drugs. For example, the word "Throat" has a much higher weight in classifying a drug as cough medication than the word "Aches" does. For the results of our Principal Component Analysis please refer to our appendix. In our PCA analysis, we found the word "Nose" to have the highest weight in PCA 1 and the word "Cold" to have the highest weight in PCA 2. The graph above shows the breakdown of the percentage of occurrences of each word in each category. From a logical standpoint, "Nose" would be a good variable to distinguish between categories because "Nose" was only found in 3 of 5 categories rather than all 5.

To run supervised data analysis, our data frame had to be split into testing and training sets. We used the nitial_split(), training() and testing() functions to create our splits along with the 5 fold cross-validation. We tried many model types such as decision trees, random forests and neural networks using the TidyModels framework. In the end, our neural network had the highest accuracy rate and therefore that is the model that we chose to use. A classification recipe, workflow, and grid were created and used to tune hidden units and penalties for our neural network. The best accuracy was produced with Hidden Units set at 7 and Penalty set at 1. These metrics created our final neural network model which can be seen in the appendix. This neural network had an accuracy rating of 61%, refer to figure 3, and concluded the first layer of our classification tree. To better understand how strong our model's classification accuracy is one must understand that a random guess would only be correct about 20% of the time.

We followed a similar process for the second layer of our classification tree. Using the same cleaned dataset, we ran a function that created a dummy variable indicating whether the brand name was present in the drug's name. For example, a drug would be classified as "brand-name" if the drug name was "Robitussin Cough + Congestion DM Max Syrup" and the corresponding brand was "Robitussin". And likewise, a drug would be classified as "generic" if the drug name was "Daytime Cold & Flu Relief Softgels" and the brand was "Up & Up".

For the train set of this data we had to manually put in whether our drugs were actually brand-name or generic, to test our hypothesis. Afterwards, we were able to run our classification model. The model simply produced whether the drug was generic or brand-name.

The final stage of our project was to make all of our code accessible and usable to the World Bank. To do this, we wanted to automate as many steps as possible by creating functions. Because the World Bank would be pulling drug data from all around the globe and from all types of stores, we could not automate the cleaning process. Depending on the country and store, scraped data could look completely different. We did create a general guideline for how scraped data needed to look to run on our models. We aimed to make the necessary steps as doable as

possible. The most important component of making data readable by our models was to make sure that there was a column for name, brand, uses, and price range. The columns needed to be labeled correctly in order for our models to recognize them.

We created a Shiny interface in R that allowed the World Bank to submit their data and have the classification at both the category and brand versus generic levels to be classified automatically. The interface would produce the predictions in a readable tabular form. We tested the entire process on pharmaceutical data scraped from Rite Aid's website. To test the accuracy, we scraped only cough medications. In doing so, we found that our models had a 60% accuracy rate.

Going forward, there are many steps that we believe the World Bank could take in order to take advantage of and improve our models. The first step would be to get more training data. Because we used ParseHub, a free scraping tool, we were limited in the number of pages that we could scrape at one time. With more training data, from more sources, the models could become exponentially more accurate and representative of a larger range of pharmaceuticals. Going hand in hand with the previous step, it would be beneficial for the World Bank to use a different web scraping tool. ParseHub is great for people who do not have prior experience with scraping but it has size limitations and takes a lot of time and computing power. The final step that the World Bank could take would be to implement cloud services into the process. The cloud offers more computing power as well as many useful tools such as translators that could make global data more accessible.

Overall, we are thankful for the opportunity to work with the World Bank on such an exciting project. We are very proud of what we were able to accomplish in a short period of time and hope that the World Bank finds our work useful and interesting.

Appendix

Figure 1: Principal Component Analysis Weights

Figure 2: Frequencies of the Top 35 Words

This graph depicts the frequencies within our dataset of all 35 words we used as dummy variables to predict on. The color coordinates to what percentage of the occurrences are attributed to each category.

Figure 3: Accuracy