THE WORLD BANK

International Comparison Progam: Household Product Classification Final Report

Nick Bias
Will Gushurst
Nolan Neel
Joseph Willemsz

California Polytechnic State University
MS Business Analytics

Winter 2022

Table of Contents

Introduction

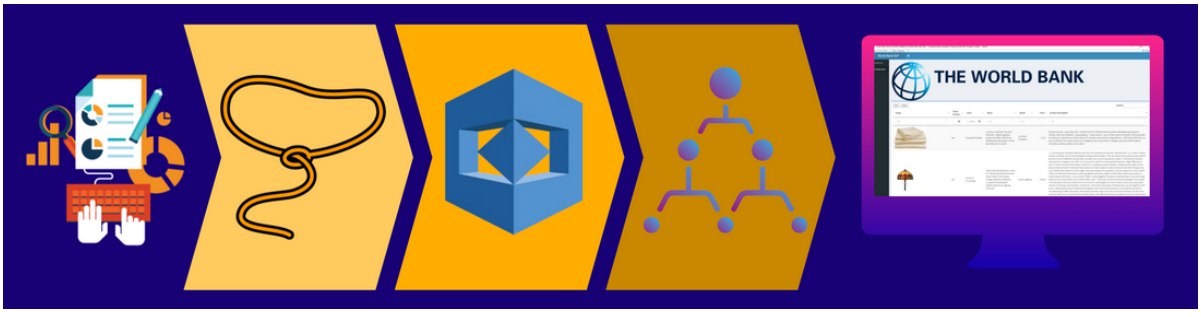
The International Comparison Program (ICP) has made a lot of progress toward analyzing purchasing powers between countries by creating an international comparison index, by which similar products get classified and compared using structured product definitions. These categories range from foods and household products to common services such as a haircut. Many products in each country are grouped together to make up a comparable basket of goods that can be analyzed to compare relative prices among countries around the world to further understand purchasing power parities. The areas for innovation presented to us at the beginning of the quarter included the use of natural language processing (NLP) as a means for classifying products, and the challenge of finding the most efficient way to classify products into their respective ICP categories as defined by The World Bank.

Our Approach

Using the World Bank's guidance in our initial meetings along with the materials provided to us, we got started on our goal to find the most efficient way to accurately classify products down the classification funnel from main aggregate to basic heading. The dataset we used as the basis for this project was full of products that were scraped from Amazon's website. We found that a majority of these products were household products and fit into the ICP classes rather nicely. In order to do this efficiently, we created the outline for a pipeline that could take a dataset with the proper information and ultimately output a list of products with their predicted ICP classes. The use of this pipeline would assist in automating an otherwise difficult classification process, thus allowing for more consistent routine releases of international comparisons and purchasing power parities that reflect the most current data possible.

In addition to text data, our dataset also contained image links for each product. This presented an excellent opportunity to utilize new techniques for analyzing the data that may not have been available to us otherwise. These image links allowed us to harness cloud computing capabilities through Amazon Web Services (AWS). Within AWS we used a service called Amazon Rekognition to take advantage of its image recognition capabilities. Amazon Rekognition has a key feature named Custom Labels, which allowed us to create an image classification model based on specific label categories. Ultimately, we found that using Amazon Rekognition to predict ICP classes was an excellent indicator when the prediction was used as a variable in our final decision tree model.

Our team made an effort to implement the recommendations and notes given to us during our midpoint review, and turning our proposed process into a streamlined pipeline was a hope that was emphasized by The World Bank. To implement this pipeline, we started with a lasso model, which decides whether an object is a household product or not. Next, we used our Custom Labels model in Amazon Rekognition to predict the ICP class for each product. From here, the predictions are used as input for a final decision tree model, in which products are classified into their ICP categories. Finally, we've created an interface that ties this all together to make it streamlined and interactive.



Model 1: Lasso

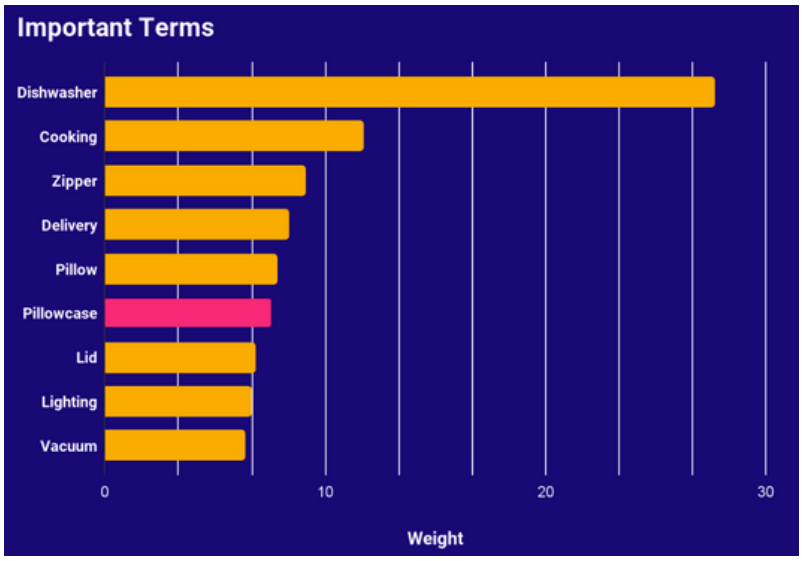
When looking at the data for household products, we found that most products have a few similar points of data that we could use to identify the product: Name, Brand, and Product Description. The model that we found that best used this data was a logistic model with lasso regression.

Data Cleaning

Before creating the lasso model, we determined the specified ICP class for each of the observations within Microsoft Excel. Here we manually entered the ICP class category within a new column named icp_class for each of the observations in the dataset. Although there are twelve classes within the "furnishings, household equipment, and routine household maintenance" category, we used seven of those twelve classes since our data primarily included products that fell into those classes. Products in our dataset that were labeled as a household class were then also marked as "Household" products.

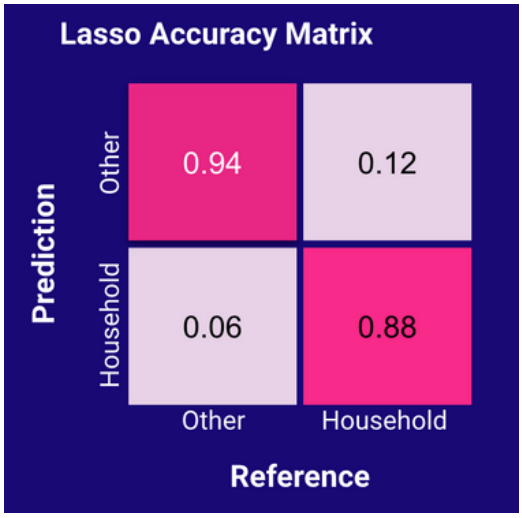
Model

The lasso model that we created is a binary classification model that uses NLP to classify products as a household product or something else. This model uses tokenization to break apart the text in the Name, Brand, and Product Description fields and create variables for each word. Once variables are created, the lasso model penalizes certain words to either lower their weight or remove them from the model. As a result, common words such as "the" or "and" are no longer considered important to the model due to their high frequency. The graph below depicts the most important words that the lasso model considers when predicting if a product is a household object or not. Here we can see that the word dishwasher in the product description is considered very important to our model.



Results

When the model is used to predict on our entire dataset, it predicts with 91.64% accuracy. A heat map of the results can be seen in the chart to the right. The value in each cell is the percentage of the reference category within the cell. So in the top left cell, the 0.94 indicates that 94% of products that were considered part of the "Other" category were classified by the lasso model as an "Other" product. Further details about this model and its accuracy can be viewed in the appendix.



Model 2: Initial Decision Tree

In addition to the Amazon Rekognition Custom Labels feature, we wanted to continue our use of NLP to predict a product's ICP class. This second model takes in similar inputs as our Lasso model and uses them to predict if the product fits into one of the following ICP categories:

- Furniture and Furnishings
- Glassware, Tableware, and Household Utensils
- Household Textiles
- Major Household Appliances
- Non-Durable Household Goods
- Small Electric Household Appliances
- Small Tools and Miscellaneous Accessories

We decided to use a decision tree since tokens were a good input for the decision tree splits and the model had relatively high accuracy. Since the input for the decision tree was the same as the lasso model, no data cleaning was necessary for this step.

Results

When evaluating the model, we found that it had an overall accuracy of 82%, which while still relatively high, wasn't at the same level as our Lasso model for making these classifications. A heat map of the results can be seen below.

	Reference						
Prediction	FF	GTHU	HT	MHA	NHG	SEHA	STMA
Furniture & Furnishings	77%	5%	3%	9%	3%	22%	7%
Glassware, Tableware, Household Utensils	15%	86%	5%	41%	31%	42%	18%
Household Textiles	6%	6%	92%	3%	2%	4%	2%
Major Household Appliances	0%	0%	0%	44%	0%	1%	0%
Non-durable Household Goods	0%	2%	0%	0%	65%	7%	0%
Small Electric Household Appliances	0%	0%	0%	1%	0%	24%	0%
Small Tools & Misc Accessories	1%	0%	0%	2%	0%	0%	74%

The value in each cell is the percentage of the reference category within the cell, similar to the Lasso model's chart. While in some categories the model does fairly well, like "Household Textiles" or "Glassware, Tableware, and Household Utensils", in others the accuracy drops drastically, with the model only classifying 24% of "Small Electric Household Appliances" correctly. In order to improve our predicting power, we turned to Amazon Rekognition to see if a model using image recognition would improve upon these results.

Amazon Rekognition

Model Preparation

In order to train a Custom Labels model in Amazon Rekognition, we had to have a set of images that were pre-labeled. At this point we only had a dataset containing the URLs for each observation and not the actual image files. In order to download the images necessary to train the Rekognition model, we used the updated dataset that contained the icp_class variable and imported this into R. Next, we created an automated process to download the associated .jpg file for each url into specific folders corresponding to the various ICP home product class categories by filtering through the icp_class variable. Placing these images in certain folders pre labels the images. After all the images were downloaded into their correct class folders, the image folders were uploaded into an Amazon S3 bucket, an AWS Cloud Storage Service, ready to be passed into Amazon Rekognition where the model is trained.

Results

After training our Amazon Rekognition Custom Labels model, we received model evaluation results of an average F1 score of 0.772, average precision of 0.814, and overall recall of 0.747. There was also a performance breakdown for each of the labels. Some labels had higher performance scores since these classes were less broad than others and contained more training images. For example, the Rekognition model was successful in predicting the "Household Textiles" class with an F1 score of 0.936 since this product class contained similar products with many images. On the other hand, the "Small Electric Household Appliances" class had a lower F1 score of 0.612 since this class had less training images and a larger variety of products to train on.

Predictions

The next step in Rekognition was deploying our Custom Labels model to make predictions and output the ICP class for each home product in our dataset. When making predictions, a label along with a confidence level is outputted for each of the predictions. For each prediction we received each of the seven class labels along with an associated confidence level. The prediction output was given in a JSON format that was cleaned using the pandas library within Python in order to be used in the upcoming decision tree model. After cleaning the Rekognition output we wrote this data frame to a CSV file that was later used to run a final decision tree classification model.

Final Model: Decision Tree with AWS Rekognition Integration

Since Amazon Rekognition did not perform significantly better than our decision tree, we decided to combine these models to try and achieve better results. The final model we created was still a decision tree, but in addition to the text used for NLP, we added the results from Rekognition. Seven new variables were added to the model, which were Rekognition's confidence that an item fit into a certain category. For example, Rekognition could be 50% certain that a tapestry is in Household Textiles, 50% certain that it is in Furniture and Furnishings, and 0% confident that it is in any other category.

Combining the models in this fashion results in noticeable improvements to model accuracy. The accuracy jumps from our previous 82% to 92% accurate when making predictions on ICP class. A heat map of the new results is shown below:

	Reference						
Prediction	FF	GTHU	HT	MHA	NHG	SEHA	STMA
Furniture & Furnishings	90%	4%	1%	6%	2%	6%	8%
Glassware, Tableware, Household Utensils	7%	93%	2%	11%	13%	18%	4%
Household Textiles	2%	1%	97%	0%	1%	1%	0%
Major Household Appliances	0%	0%	0%	80%	0%	0%	0%
Non-durable Household Goods	0%	1%	0%	1%	83%	5%	0%
Small Electric Household Appliances	0%	0%	0%	1%	0%	71%	0%
Small Tools & Misc Accessories	1%	0%	0%	0%	1%	0%	87%

There is noticeable improvement to the heat map with most of the color being along the diagonal, indicating that the results are more accurate. Results in all categories were improved with the addition, including "Household Textiles" which was improved to 97% accurate. Our worst category from the NLP model was "Small Electric Household Appliances" and the accuracy from that category has been dramatically improved, rising from 24% accurate to 71% accurate.

Shiny App

To implement a final deliverable for the World Bank, we designed an interface that would allow the clients to visually compare items together to see if they are comparable or not. This was done with the use of Shiny, which is an R package that allows a user to make interactive web apps that can be hosted. Throughout the course of these 10 weeks, we created many iterations of the interface, each time adding a new feature that the client might appreciate.

The final interface is a dashboard with only two tabs. The first tab is called “Data Input”. As the name implies, this is where the user uploads the dataset. For the classification models to work, there are five required variables. These variables are the product name, the product brand, the product description, and a URL for the product image. If these variables are not named 'product_image_url', 'name', 'brand', 'product_description', and 'price' then the classification will not work. Price is technically not needed to run classification, but we decided that it would be helpful for the clients to be able to look at the price when comparing the products.

The initial supplied dataset is run through our lasso model first. Here it determines if the item is a household product or not. Then just the product image URLs are sent to a Python file, where AWS Rekognition is utilized to classify the product categories, according to the ICP home categories. This outputs a file that is then joined with the original data. This new dataset is then run through our final decision tree model and does the main classification. Finally, the classification table results are displayed in the second tab of the dashboard, titled “Classification.” In each row the product image is displayed first, followed by whether the product is a home product or not, the classification category, and then the name, brand, description and price of the product. From here the user can easily see what the product is and how it was classified by the model. The user can easily filter by any variable in the dataset. For items that get misclassified by the model, the user can simply double-click on the category and reclassify the product themselves. Once the user has a view of the dataset they are satisfied with, they can click either the “CSV” or “Excel” button at the top of the table, to save the file in the format of their choice.

While the interface presents a promising proof of concept, there are certainly opportunities for improvement. When a dataset is uploaded to the interface, it takes a few minutes to be able to view the classification results. The time it takes to run classification varies depending on how large the supplied dataset is. Occasionally the code runs into errors; however, this is often due to a problem in the reticulate package used to combine two programming languages (R and Python) into the same package. These are errors that can be worked out and troubleshooted with ease.

Conclusion

To effectively implement this into your program, we would recommend training this model on a dataset that has been thoroughly cleaned and expanded to other ICP categories outside of household products. While these models were built utilizing the data we had available to us, we believe this process is scalable to many other categories within the ICP’s classifications. Our team is hopeful that using the models and the interactive app we’ve created will allow for a user to compare products in an easier, more streamlined process.

Appendix

Lasso Model confusion matrix output.
Relevant Statistics are included.

```
Confusion Matrix and Statistics


       0     1
  0 29704  2834
  1  1787 20952

               Accuracy : 0.9164
                 95% CI : (0.9141, 0.9187)
    No Information Rate : 0.5697
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8286

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.8809
            Specificity : 0.9433
         Pos Pred Value : 0.9214
         Neg Pred Value : 0.9129
             Prevalence : 0.4303
         Detection Rate : 0.3790
   Detection Prevalence : 0.4114
      Balanced Accuracy : 0.9121

       'Positive' Class : 1
```

Legend for the two following confusion matrix outputs.

icp_class <fctr>	class_id <fctr>
furniture_furnishings	1
glassware_tableware_household_utensils	2
household_textiles	3
major_household_appliances	4
nondurable_household_goods	5
small_electric_household_appliances	6
small_tools_misc_accessories	7

Appendix

Initial decision tree prior to the inclusion of Amazon Rekognition output

```
Confusion Matrix and Statistics


          1    2    3    4    5    6    7
  1 4608  335  270   36   33  107   69
  2  913 5317  416  170  339  199  177
  3  378  354 7969   14   18   19   19
  4    3    8    0  179    0    3    2
  5    0  136    4    1  713   35    0
  6    0   19    0    3    0  113    0
  7   66   12    2    8    2    2  741

Overall Statistics

               Accuracy : 0.8248
                 95% CI : (0.8199, 0.8296)
    No Information Rate : 0.3637
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.7577

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6 Class: 7
Sensitivity            0.7721   0.8602   0.9201 0.435523  0.64525 0.236402  0.73512
Specificity            0.9524   0.8744   0.9471 0.999316  0.99225 0.999057  0.99597
Pos Pred Value         0.8443   0.7060   0.9086 0.917949  0.80202 0.837037  0.88956
Neg Pred Value         0.9259   0.9469   0.9540 0.990177  0.98290 0.984584  0.98838
Prevalence             0.2506   0.2596   0.3637 0.017260  0.04641 0.020074  0.04233
Detection Rate         0.1935   0.2233   0.3347 0.007517  0.02994 0.004746  0.03112
Detection Prevalence   0.2292   0.3163   0.3683 0.008189  0.03733 0.005669  0.03498
Balanced Accuracy      0.8622   0.8673   0.9336 0.717420  0.81875 0.617729  0.86554
```

Final decision tree with the inclusion of Amazon Rekognition category predictions

```
Confusion Matrix and Statistics


          1    2    3    4    5    6    7
  1 5379  255   99   26   26   29   85
  2  401 5730  160   47  143   84   43
  3  100   63 8394    0    8    3    5
  4   19   21    0  328    1    0    1
  5    9   76    7    3  912   25    0
  6   20   14    1    6    0  337    0
  7   40   22    0    1   15    0  874

Overall Statistics

               Accuracy : 0.922
                 95% CI : (0.9185, 0.9253)
    No Information Rate : 0.3637
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8932

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4 Class: 5 Class: 6 Class: 7
Sensitivity            0.9013   0.9270   0.9692  0.79805  0.82534  0.70502  0.86706
Specificity            0.9709   0.9502   0.9882  0.99821  0.99472  0.99824  0.99658
Pos Pred Value         0.9118   0.8671   0.9791  0.88649  0.88372  0.89153  0.91807
Neg Pred Value         0.9671   0.9738   0.9825  0.99646  0.99153  0.99398  0.99414
Prevalence             0.2506   0.2596   0.3637  0.01726  0.04641  0.02007  0.04233
Detection Rate         0.2259   0.2406   0.3525  0.01377  0.03830  0.01415  0.03670
Detection Prevalence   0.2477   0.2775   0.3600  0.01554  0.04334  0.01587  0.03998
Balanced Accuracy      0.9361   0.9386   0.9787  0.89813  0.91003  0.85163  0.93182
```

Appendix

Amazon Rekognition Custom Labels model evaluation results

Evaluation results — View test results

F1 score Info	Average precision Info	Overall recall Info
0.772	0.814	0.747
Date completed	**Training dataset**	**Testing dataset**
February 19, 2022 Trained in 4.444 hours	7 labels, 8,877 images	7 labels, 2,221 images

Per label performance (7)

Find labels — 1

Label name	F1 score	Test images	Precision	Recall	Assumed threshold
furniture_furnishings	0.826	406	0.833	0.820	0.390
glassware_tableware_household_utensils	0.790	665	0.724	0.871	0.206
household_textiles	0.936	580	0.929	0.943	0.500
major_household_appliances	0.701	82	0.904	0.573	0.800
nondurable_household_goods	0.722	196	0.735	0.709	0.344
small_electric_household_appliances	0.612	86	0.738	0.523	0.832
small_tools_misc_accessories	0.813	206	0.836	0.791	0.533

Example output data frame from Amazon Rekognition

	A	B	C	D	E	F	G	H	I
1	url	furniture_furnishings	glassware_tableware_housel	household_textiles	major_household_applianc	ondurable_housel	small_electric_hc	small_tools_misc_accessories	
2	https://i5.walma	0.001	0.510999978	0.015000001	0.75	97.0759964	1.644000053	0.004	
3	https://i5.walma	0.640999973	63.11100006	0.284999996	11.96099949	19.89499855	3.911000013	0.194999993	
4	https://i5.walma	0.001	2.109000206	0	0.009000001	0.215999991	97.66600037	0	
5	https://i5.walma	0.048	69.93499756	0.743999958	0.681999981	27.89599991	0.052000001	0.643999994	
6	https://i5.walma	0.329999983	64.69400024	0.675000012	22.20299911	7.852000237	4.06000042	0.18599999	
7	https://i5.walma	0.280999988	1.498999953	0.133000001	90.71499634	6.128000259	0.856000006	0.388000011	
8	https://i5.walma	0.088	0.467000008	1.696000099	97.60400391	0.064999998	0.013	0.066	
9	https://i5.walma	0.152999997	31.56999969	0.07	21.29700089	4.013999939	22.98099899	19.91399956	
10	https://i5.walma	0.338999987	91.09100342	0.313999981	0.007999999	7.749000072	0.450999975	0.049000002	
11	https://i5.walma	0.271999985	22.14400101	0.07	8.18500042	40.3390007	25.74099922	3.249000072	
12	https://i5.walma	0.002	0.526000023	0.005	0.394999981	99.01900482	0.041999999	0.012	
13	https://i5.walma	1.376999974	1.286000013	0.54399997	53.76300049	0.18599999	0.384000003	42.45899963	