Food Product Prices across Portugal's biggest supermarket chains

André Oliveira (20222156@novaims.unl.pt, 20222156); Rafael Chamusca (20222162@novaims.unl.pt, 20222174); Diogo (20222152@novaims.unl.pt, 20222152); Israel Lucena (20211720@novaums.unl.pt, 20211720)

Group. B3 TP 2

Abstract

This project's background goal is to collect the prices of the 25 items included in the Food Baskets: Operational Program to Support the Neediest People using python. To extract the data, we used the libraries Selenium, Requests, Beautiful Soup and lxml. To clean and filter we used traditional data cleaning techniques and the library rapidfuzz. We filter the data set of the pingo doce and continente groceries products information with 16866 observations and 8 features two times. Once to a data set with 3639 observations for analysis purposes. And a second to a data set with 1236 observations to make it possible to inform the consumer of the products available in both supermarkets considering the 25 Food Baskets. There is fierce price competition between the two retailers, along fixed category of food products. The median price of each category is very close.

Keywords. Web scraping; data cleaning; backdoors; fuzzy ratio

Introduction

Web scraping of online retailers is a non-trivial issue. In the context of rising food prices, we want to determine if two of the largest food retailers are really competing in the price dimensions or if, conversely, they have different price ranges for the same products. By examining the pricing policies of Continente and Pingo Doce we can get significant knowledge and see how prices differ between supermarkets and different categories. By finding the best bargains among retailers, our analysis can assist customers in saving money. Our approach to this project follows the following Data Science principles: **Data Collection**: We employ web scraping techniques to extract product data, including categories, prices, and weight directly from Pingo Doce and Continente's websites. This process enables us to gather a dataset that reflects the current product offerings and their prices. **Data Cleaning**: Once the data is collected and organized into a tabular format, we perform data cleaning tasks. This involves extracting variables from the string format entries, ensuring a standardized format for each variable in our final data frame. We address missing and duplicate values and employ methods to standardize the data, considering the inherent dissimilarities between the web-scraped data from different websites. **Data Visualization and Analysis**: With the data cleaned and organized, we leverage data visualization techniques to gain insights. By plotting the data, we visually represent the distributions of thousands of products per category and supermarkets. This visualization allows us to examine the distribution of overall prices, compare categories, and conduct exploratory data analysis.

Data and Methods

To acquire the data, we used the Requests and Selenium libraries. To parse the data, we used Beautiful Soup and lxml. In the case of Pingo Doce, the webpage was dynamically loaded with JavaScript. Requests alone couldn't do the job. We finally found a pattern that we could use. To retrieve the text information, we used Beautiful Soup. In the case of Continente the most used libraries were BeautifulSoup and Requests. We divided the process into 3 steps. In the first, we extracted the information contained on the home page focusing on the links referring to the categories of products. We selected 9 categories for the next phase. In step 2, we went through the pages every 24 products per page, repeating the orders and increasing the pagination index. We then had the response to requests for each page of the respective category. In step 3, the final analysis was performed to extract relevant information from each product. BeautifulSoup was used to facilitate the indexing of existing values in the div, span and title tags, as well as filtering by the class attribute existing in the HTML tags. Lastly, we got all the products in a dictionary which was concatenated by the 9 categories and turned into a dataset for further analysis. After converting these data into tabular format, we noticed similar structures, and the main characteristic of these datasets is a column with important information in a string format, detailing the weight per unit and price per weight. Thus, we manipulated the column to create the features price, amount and units. Pingo Doce's dataset initially had dimensions of (6226, 5) and after the data cleaning process it became with (5601, 8). Continente's initially had dimensions of (12251, 12) and after data cleaning it became (11265, 8). We cleaned and concatenated both datasets for further analysis and exploration. To analyze the Food Baskets we filtered the dataset 2 times. In the second time we used the rapidfuzz library. A fast string matching library which uses the string similarity from the FuzzyWuzzy library. The logic behind the 2 filters is the same, but the method process of the rapidfuzz lib allows us to improve the accuracy of the filter.

Table 1. Conversion table between categories of the food basket

Category	Pingo Doce	Continente	Abbreviation
mercearia	mercearia	mercearia, bio-e-escolhas-alimentares'	MERC
padaria_pastelaria	padaria_pastelaria	padaria-e-pastelaria	PAPA
frutas_legumes	frutas_legumes	frutas_legumes	FRLE
peixaria_talho	talho, peixaria	peixaria-e-talho	PETA
leite_ovos	leite_ovos_natas	laticinios-e-ovos	LEOV
congelados	congelados	congelados	CONGE
frigorífico	take-away, frigorífico	refeicoes-faceis, charcutaria-e-queijos	FRIGO

Results and Discussion

A larger dataset, with 3,629 prices, was used to compare Continente (2,413 products; 66,5%) and Pingo Doce (1,216 products; 33,5%). The highest price is 35.47€, the lowest is 0.15. From Figure 1, we can see that MERC is the category with the highest prices, while the lowest prices are in PAPA. Figure 2 tells us that the distribution of prices in each category is heavily skewed to the right, with a thin tail. In other words, unit prices are clustered near zero, with some large outliers (. The same pattern is repeated in Figure 3, where the prices of both retailers are clustered around small values, with very large outliers. These results show a fierce competition on the price dimension.

Figure 1. Range of Prices per Category

Figure 2. Histogram of Prices per Category

Figure 3. Histogram of Prices per Store

Figure 4 shows that the median price is higher in Continente than in Pingo Doce in five out of seven categories. Overall, Continente has the higher median price (PETA), while Pingo Doce has the lower median price (LEOV). We use the median instead of the mean price given the asymmetry in the distribution of prices.

Figure 4. Median prices per category in each store

We now move to a statistical method to enable us to uncover the variance structure of the dataset. Since we have both numerical and categorical variables, we chose Factor Analysis of Mixed Data (FAMD). FAMD works as Principal Component Analysis (PCA) for numerical data and Multiple Correspondence Analysis (MCA) for categorical data. Graphically, the cloud of points is fully represented on the plane formed by the intersection of the two main components. The variables are also represented by their centers.Our price variable was not normally distributed. Since normality is required to apply FAMD, we subjected List_Price to the Yoe-Johnson (Power) transformation. The result was successful. The model, with two principal components, explains 31,7% of the variance, which is not much, but suffices to our goal. The variables more associated with the first component (C0) are Price and Category, while Store is associated with the second component (C1). Figure 5 shows that C0 represents price, which moves rightwards – the highest prices are on the far right and the lowest on the far left. C1 perfectly separates the retailers – all Pingo Doce's prices are above the 0.5 mark on the ordinate axis; all Continente's prices are below. The leftmost green square is Store, placed slightly above the 0.5 line. Visual inspection of Figure 5 confirms our findings. Continente has slightly higher prices, although the price structure of both retailers is similar.

Figure 5. Plot of principal components in FAMD

Conclusions

There is fierce price competition between the two retailers, along fixed category of food products. The median price of each category is very close. This is very hard to achieve since there are thousands of products. The dataset in which we based our analysis has 3,629 products and only encompasses the food categories. Such closeness of prices is only possible if both retailers are continually adjusting their prices, to keep them on a very narrow interval. Given the number of prices involved this is only possible through constant monitoring and web scraping between them. Hence the barriers we faced.

References

Beautiful Soup Documentation. (n.d.). Beautiful Soup 4. Retrieved May 21, 2023, from https://beautiful-soup-4.readthedocs.io/

Hillen, J. (2019). Web scraping for food price research. *British Food Journal*, *121*(12), 3350-3361.

Manjushree, B. S., & Sharvani, G. S. (2020). Survey on Web scraping technology. *Wutan Huatan Jisuan Jishu*, *16*(6), 1-8.

Mehak, S., Zafar, R., Aslam, S., & Bhatti, S. M. (2019, January). Exploiting filtering approach with web scrapping for smart online shopping: Penny wise: A wise tool for online shopping. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-5). IEEE.

RapidFuzz Documentation (n.d.). Rapid fuzzy string matching in Python and C++ using the Levenshtein Distance. Retrieved May 21, 2023, from https://pypi.org/project/rapidfuzz/

Requests Documentation. (n.d.). Requests 2.26.0 documentation. Retrieved May 21, 2023, from https://requests.readthedocs.io/en/latest/

Scrapy Documentation. (n.d.). Scrapy 2.5.1 documentation. Retrieved May 21, 2023, from https://docs.scrapy.org/en/latest/

Selenium-Python Documentation. (n.d.). Selenium with Python 4.1.0 documentation. Retrieved May 21, 2023, from https://selenium-python.readthedocs.io/