Amazon Cell Phone Reviews

Milestone I

André Gomes
FEUP
up201806224@edu.fe.up.pt

André Nascimento
FEUP
up201806461@edu.fe.up.pt

Catarina Fernandes
FEUP
up201806610@edu.fe.up.pt

ABSTRACT

For the first milestone, an initial search was made to find a suitable dataset to be used for the rest of the project. After settling with a dataset about Amazon cell phones reviews retrieved from Kaggle, there was an initial data analysis of the data, followed by a scraping of updated data. A merge of updated and already existing data was made to have a more complete dataset, followed by a data exploration using OpenRefine facets and Python's Pandas library. Afterwards, the data was cleaned and refined process using Python scripts, and, to finalize the pipeline, sql files to create and seed the database, and some graphs where created.

KEYWORDS

scraping, data retrieval, data processing, Amazon cell phone reviews

1 INTRODUCTION

Information communication technologies are ubiquitous in modern societies. It is a broad subject that covers any product that will store, retrieve, manipulate, transmit, or receive information electronically in a digital form [3]. An ever-growing number of activities depend on the ability to extract value from information and Human progress and welfare is largely dependent on an efficient management of the life cycle of information.

With this rise in the importance of data and its processing, new professional profiles have been appearing, such as: data engineer, data architect, data analyst, data scientist.

In this course, we were subjected to an introduction to information processing and to information retrieval. To solidify our knowledge, we were tasked with developing an information search system, including work on data collection and preparation, information querying and retrieval, and retrieval evaluation.

The project is divided into three milestones, namely: data preparation, information retrieval, and final search system.

This report covers the first milestone, data preparation, in which we performed the following actions:

- search repositories for datasets;
- select convenient data subsets;
- assess the authority of the data source and data quality;
- perform exploratory data analysis;
- prepare and document a data processing pipeline;
- characterize the datasets, identifying and describing some of their properties;
- identify the conceptual model for the data domain;
- identify follow-up information needs in the data domain.

2 DATA COLLECTION

While searching for datasets, we faced common problems amongst the data, such as: low amount of rows and/or columns, low word count in each entry for a given column (we need columns with large bodies of text to later perform full text search) and not so diversified text bodies (we were looking for datasets which weren't too repetitive).

The first dataset that caught our eyes was related to wine reviews. It consisted of a single .csv file with 130.000 entries. Each entry had information about a bottle of wine and its review. We could gather information about the wine, like its designation, country of origin, region, province and price, and information about the taster and its review, like the taster's name, his classification of the wine and his description of the taste.

We ended up ditching this first dataset, not because it fell under any of the common problems mentioned earlier, but because we thought we did not have enough information to work with, since this dataset only gave us one table to manage.

Later on we found the dataset explored in this report: Amazon cell phone reviews [6]. We found it via Kaggle [1], a platform used to host data science competitions and explore, analyse and share datasets. This dataset is composed of two .csv files: a list of cell phones and a list of cell phone reviews. The data was obtained from amazon via scraping by Griko Nibras, the author of the dataset and the scraping code, which wanted to aggregate the amazon reviews of cell phones from 10 brands of his choice: Asus, Apple, Google, HUAWEI, Motorola, Nokia, OnePlus, Samsung, Sony, and Xiaomi.

This dataset was obtained from amazon in 2019, making it a quite recent aggregate of data, a good point that inclined us to use it in this project.

From the first .csv file, the phones list, we can retrieve information about various phones brands, the titles of the amazon listing, the amazon url, the main image url and the url of the reviews, as well as some metrics like the original price and current price of the device, the total number of reviews and the rating of the product, from one to five. This file contains 720 entries.

From the second .csv file, the reviews list, we have various reviews of the cell phones, which include information about the reviewer, like its name and its validity (a amazon metric), and information about the review: its title and body of text, the date of the submission of the review, the score given to the product and the amount of "helpful votes" of that review, more commonly known as upvotes, which is how many people signed that review as helpful. This file contains 68 thousand entries.

Griko Nibras was generous enough to provide the code with which he performed the scraping [5], and since the data is at least two years old, we decided to get another set of .csv files by running the scraping ourselves. The downside of this approach was that

Supervised by Sara Fernandes.

Information Processing and Retrieval,
2021.

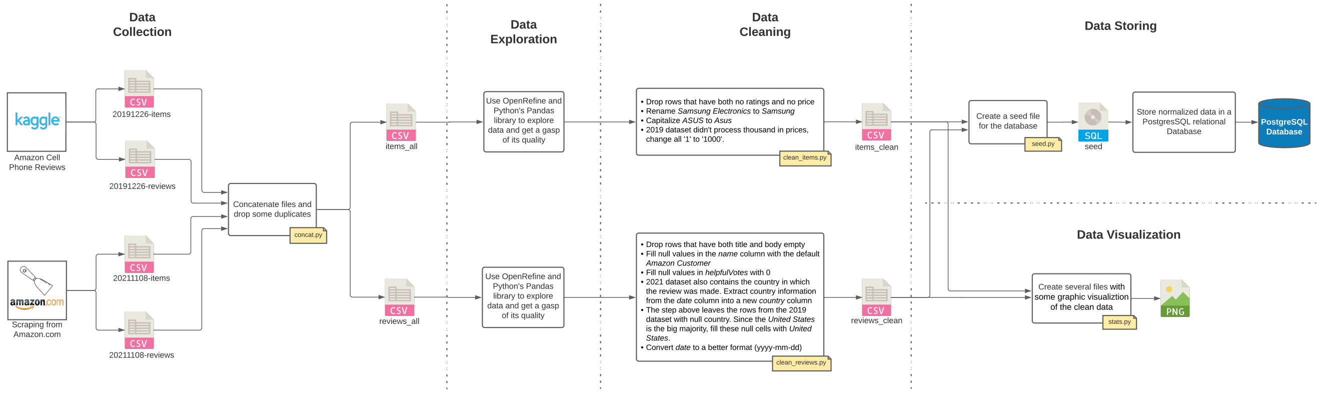
Data Collection
Data Exploration
Data Cleaning
Data Storing
kaggle
Amazon Cell Phone Reviews
amazon.com
Scraping from Amazon.com
CSV
20191226-items
CSV
20191226-reviews
CSV
20211108-items
CSV
20211108-reviews
Concatenate files and drop some duplicates
concat.py
CSV
items_all
CSV
reviews_all
Use OpenRefine and Python's Pandas library to explore data and get a gasp of its quality
Use OpenRefine and Python's Pandas library to explore data and get a gasp of its quality
• Drop rows that have both no ratings and no price
• Rename *Samsung Electronics* to *Samsung*
• Capitalize *ASUS* to *Asus*
• 2019 dataset didn't process thousand in prices, change all '1' to '1000'.
clean_items.py
• Drop rows that have both title and body empty
• Fill null values in the *name* column with the default *Amazon Customer*
• Fill null values in *helpfulVotes* with 0
• 2021 dataset also contains the country in which the review was made. Extract country information from the *date* column into a new *country* column
• The step above leaves the rows from the 2019 dataset with null country. Since the *United States* is the big majority, fill these null cells with *United States*.
• Convert *date* to a better format (yyyy-mm-dd)
clean_reviews.py
CSV
items_clean
CSV
reviews_clean
Create a seed file for the database
seed.py
SQL
seed
Store normalized data in a PostgresSQL relational Database
PostgreSQL Database
Data Visualization
Create several files with some graphic visualiztion of the clean data
stats.py
PNG

Figure 1: Data Preparation Pipeline

we needed to update the initial code, this was mainly because the amazon website was updated and some things were not getting captured correctly, for example, the samsung phones are now on a category named "Samsung Electronics", the html selectors were outdated and the parsing of prices above 1000 dollars were being done incorrectly. With these small bugs fixed, we were able to perform the scraping, which took around – minutes. In the end we were able to obtain another pair of .csv files with the most updated values possible.

3 DATA LIMITATIONS

Some limitations regarding the dataset enrichment were encountered. The first one was about the reviews body of text: The scraping was performed on the english version of the amazon website, but the costumers are from around the world, as well as the reviews left on the products, in this sense, we were faced with reviews from some countries that were written in languages that contained special characters and accented letters. We decided to left these reviews in the dataset and we'll see how they fare in the future stages of the project.

Another problem we faced was that the "brand" column in the items table would sometimes be empty, not due to a scraping error, but a data error, since the product would not have a established brand selected. Luckily, most of the times, the brand name is incorporated in the title of the amazon product, which opened the possibility of filling the empty brand names with information from the title.

Another common occurrence in the data was that, in some products, the current price would be 0, which from a business stand point does not make much sense. After verifying the url's of those products, we concluded that the products with price equal to 0 were either, at the time of scraping, unavailable, for various motives, for example, out of stock, or not on sale anymore.

Lastly, there were also some items that did not have reviews, but were still being captured by the scraper, we decided to left these items just, because they can still be searched for and have all the other columns with valid data.

4 DATA PREPARATION

For a first approach, we decided to try out OpenRefine to process the dataset. After reading the official OpenRefine Documentation [4], we had a number of steps ready to perform on the data and some facets made up to analyze the data. Using the facets we were able to automatically detect the data types of some columns, find which columns have null values and convert dates to ISO-8601-compliant extended format with time in UTC: YYYY-MM-DDTHH:MM:SSZ.

With OpenRefine, we can make the data processing steps via the GUI and export a JSON file describing these steps, to be able to recreate the steps all at once without going one by one manually again. But the problem that emerged from here is that OpenRefine does not come with a CLI out-of-the-box to perform these operations. Together with the fact that OpenRefine is a server that needs to be initiated, instead of a process that can be run via a .exe or via the command line, it made the process of creating a pipeline in a Makefile more dificult than it should be. To run a pipeline using OpenRefine in a Makefile, we would need to make sure that the OpenRefine server is first running, and then we would need to find a way to send the configuration file and the data to the server. This is not specified in a simple way in the OpenRefine documentation, so we decided not to invest in OpenRefine for the processing of the data, but we still used it for its Facet's capabilities.

After dealing with OpenRefine, we decided to use python, together with the pandas library [2], for the data processing phase. We first merged the 2019 and 2021 datasets to have a more complete .csv file. In theory, the 2021 dataset would have all of the contents of the 2019 dataset, but that was not the case, we had 730 items in 2019 and 823 items in 2021, but only 88 items were on both the datasets, so by merging the two datasets, we were able to expand our data, more than we expected.

After merging, we made the following changes to the dataset in Python, starting with the Items file:

- Drop rows that have both no ratings and no price (items unavailable to buy and that have no reviews, we have no interest in these)

- Rename Samsung Electronics to Samsung, Amazon changed the brand name so the 2021 dataset contained this brand instead.
- Capitalize ASUS to Asus (some cells had ASUS and anothers Asus)
- 2019 dataset didn't process thousand in prices. Change all '1's to '1000'. Only 2 occurrences, one in the price column and another in originalPrice.

And the Reviews file:

- Drop rows that have both title and body empty (they also have no helpfulVotes)
- Fill null values in the name column with the default Amazon Customer (only 2 occurrences)
- Fill null values in helpfulVotes with 0
- 2021 dataset also contains the country in which the review was made, split date column into two columns: country and date
- The step above leaves the rows from the 2019 dataset with null country. Since the United States is the big majority, fill these null cells with United States.
- Convert date to a more data friendly format (yyyy-mm-dd)

The finalized pipeline can be seen in 1.

5 CONCEPTUAL MODEL

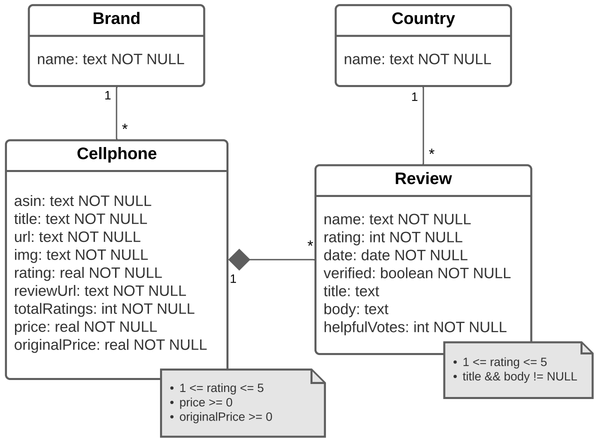

Figure 2: Conceptual Model

The conceptual model consists of a few classes, with the *Cellphone* class being the main one. The other classes help to complement it with additional information. Those classes are *Brand*, *Review* and *Country*. The main class has the following attributes that together result in the complete information of the cellphone:

- *asin* stands for Amazon Standard Identification Number. Each products has a unique asin.
- *title* corresponds to the title of the product in the publication.
- *url* is the link to the amazon products.
- *img* is a link to the main image of the product.
- *rating* is a number, from 1 to 5, corresponding to the average of all ratings.
- *reviewUrl* is a link to the first page of reviews of that item.
- *totalRatings* is the amount of ratings that the product has, each rating goes from 1 to 5.
- *price* is the current price of the product. The price could be 0, in which case, the product is not currently available.
- *originalPrice* is used when the item is in discount, this attribute would have a value and *price* would have a lower value. If the item is not on discount, this value is 0 .

The *Brand* class only has one attribute, name, which corresponds to one of the 10 brand names, as well as the *Country* class, while the *Review* class has the following attributes:

- *name* is the name of the costumer that submitted the review
- *rating* is the score, from 1 to 5, that was given to the item
- *date* is the date of when the reviews was submitted
- *country* is the country of the costumer from where he submitted the review
- *verified* is a boolean value that represents if the reviewer has bought the item
- *title* is the title of the review
- *body* is the body of text of the reviews
- *helpfulVotes* is the amount of people that marked that review as helpful

6 SEARCH TASKS

After retrieving the data, merging and cleaning, we are able to perform queries to get some insights about the dataset. Here are some of the possible queries to be made to the database:

- Search a cell phone by its title, brand, rating and price
  - Returns a list of cell phones filtered by the desired parameters
- Search a review by the reviewers name, its title, body, country
  - Returns a list of reviews filtered by the desired parameters
- Check which brands are more developed
  - Return the Number of cell phones on cell globally for each brand
- Check which products and brands are more liked by the reviewers
  - Return the number of 1/2/3/4/5 stars reviews per item and per brand
  - Return the brands with most/least positive feedback

7 DATASET CHARACTERISATION

In order to better understand and characterise the collected data, charts were developed concerning the following aspects of the database:

Regarding the cell phones data, we would like to have a visual understanding of how many cell phones are currently being sold by brand name, and we can conclude from figure 3 that, in the amazon website, Samsung has the market dominance while ASUS only has 28 cell phones for sale.

From figure 4 we can check that Xiaomi is the brand with the highest average rating, while Apple has the lowest, this may be due to the fact that most of the Apple phones that are being sold in Amazon are refurbished, and not original.

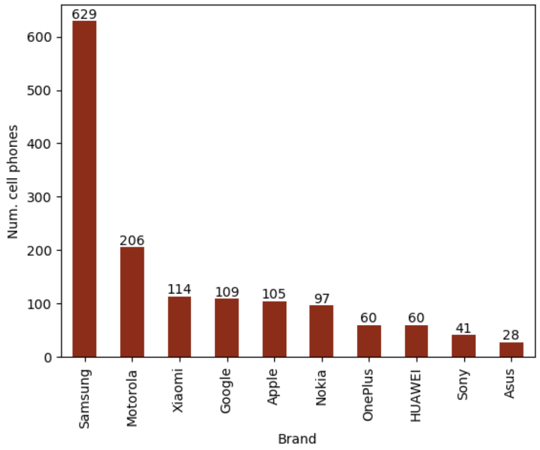

Figure 3: Amount of cell phones per brand

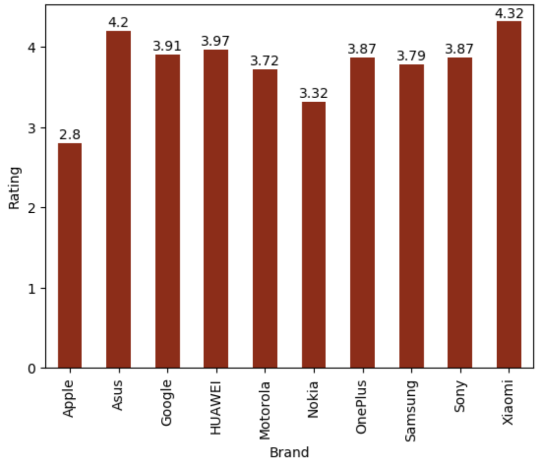

Figure 4: Average cell phone rating per brand

Concerning the reviews, from figure 5 we can conclude that the rating that is given the most is 5 stars, followed by 1, while the lowest is 2. This means that amazon users prefer to either give a 1 star rating or 5 star rating, instead of a middle ground. Since 5 stars are the norm, they are also not that valuable.

8 CONCLUSION

All of the goals for the first phase of the project were accomplished successfully given that there is a better understanding of the chosen domain, the already existent data and datasets in it and which ones are relevant for this purpose.

The only setback we had during this milestone was the data preparation using OpenRefine, that was midway changed to Python, other than that, we achieved our end goal of creating a dataset with complete and coherent data about cell phone reviews.

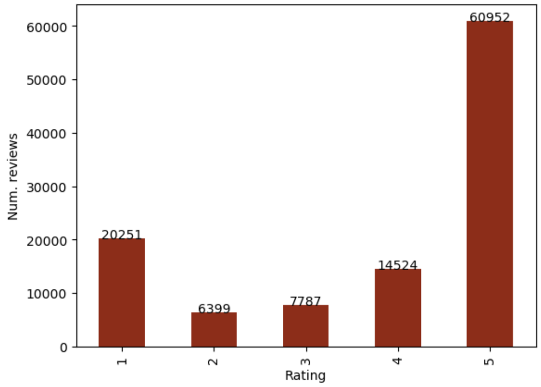

Figure 5: a nice plot

As for future work, we plan on using Solr to perform information retrieval and later on start to develop our search engine.

REFERENCES

[1] [n. d.]. Find open datasets and Machine Learning Projects. https://www.kaggle.com/datasets
[2] [n. d.]. Pandas. https://pandas.pydata.org/
[3] Information and Communications Technology. 2021. Information and Communications Technology. https://en.wikipedia.org/wiki/Information_and_communications_technology
[4] Antonin Delpeuch. 2021. Openrefine User Manual. https://docs.openrefine.org/
[5] Griko Nibras. [n. d.]. Grikomsn/amazon-cell-phones-reviews: Scrape (UN)locked cell phone ratings and reviews on Amazon. https://github.com/grikomsn/amazon-cell-phones-reviews
[6] Griko Nibras. 2019. Amazon cell phones reviews. https://www.kaggle.com/grikomsn/amazon-cell-phones-reviews