Task 0: Getting comfortable with the data
Alkemy x LUISS

Fabiana Caccavale, Matteo Gioia, Martina Manno, Vincenzo Junior Striano

Big Data and Smart Data Analytics - A.Y. 2022/2023
October 2022

Contents

1 Introduction & Methodology

The business case is about an e-commerce firm specialized in consumer electronics which sells its products both online and offline. The purpose of the case is to analyse some internal and external dynamics in order to optimize the company pricing strategy. The main goals to be achieved are:

- *Is there a First Mover in setting the price?* Understanding the pricing leaders and followers among the players in the market. This implies a deep analysis on the competitors pricing strategy and the detection of useful historical patterns and correlations.
- *What factors influence the sale of a product or product category?* Understanding whether sales are influenced by seasonality effects and extract the popularity index of products to activate targeted marketing campaigns.

To achieve these results, a database is provided, which will be analyzed below. The datasets available are as follows:

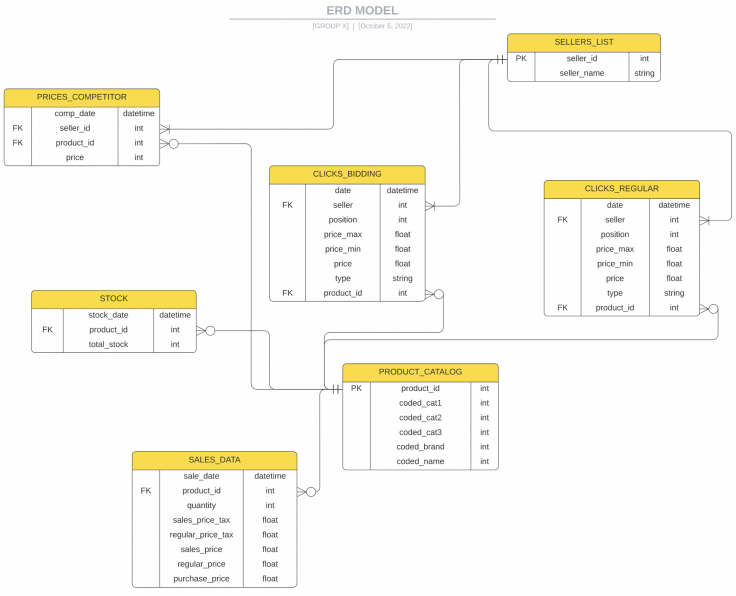
- clicks_bidding.csv
- clicks_regular.csv
- prices_competitor.csv
- product_catalog.csv
- sales_data.csv
- sellers_list.csv
- stock.csv

The aim of *Task 0* is familiarize with the data, merging them when possible, clean them, and evaluate the business requirements.

1.1 Data Integration

In order to be able to merge the datasets and gain an in-depth understanding of the relationships between the variables, an ER diagram was designed with the online tool Lucidchart. The ERD model shows what the entities are and the relationships between them. It was the starting point for understanding how the datasets could be merged.
For example, in *product_catalog*, the product_id is the primary key, or the unique identifier for a product. Whereas, in *sales_data* the product_id is a foreign key. What are the relationships?

- How many unique product_id's have been sold? Zero to many
- How many unique product_id's are in a single sales observation? One and only one



The preliminary analysis is realized using SQL implemented in Python, thanks to the *sqlite3* library. That library realizes a database, a collection of the dataset provided, and let interact with them using the SQL queries syntax. An example is provided below:

```
SELECT c.date, c.seller, c.position, c.price_max, c.price_min, c.price, c.type,
 c.product_id, p.price
FROM concatenato AS c
LEFT JOIN prices AS p
ON DATE(c.date) = p.comp_date
AND c.seller = p.seller_id
AND c.product_id = p.product_id
WHERE c.price IS NULL AND p.price IS NOT NULL
```

The datasets, the sqlite3 database creation python script, the data visualization script, and the main queries are stored into a shared, cloud-based virtual environment on replit.com for real-time edits. The next step will be to upload the major releases on a GitHub.

1.2 Data Visualization

To gain further insights and find confirmation about the findings, the Python library *plotly.express* was used for plotting the most interesting findings. Plotly was chosen thanks to its highly customizable and interactive plots.

The following sections report the results of the analysis for each single dataset.

2 Prices competitors

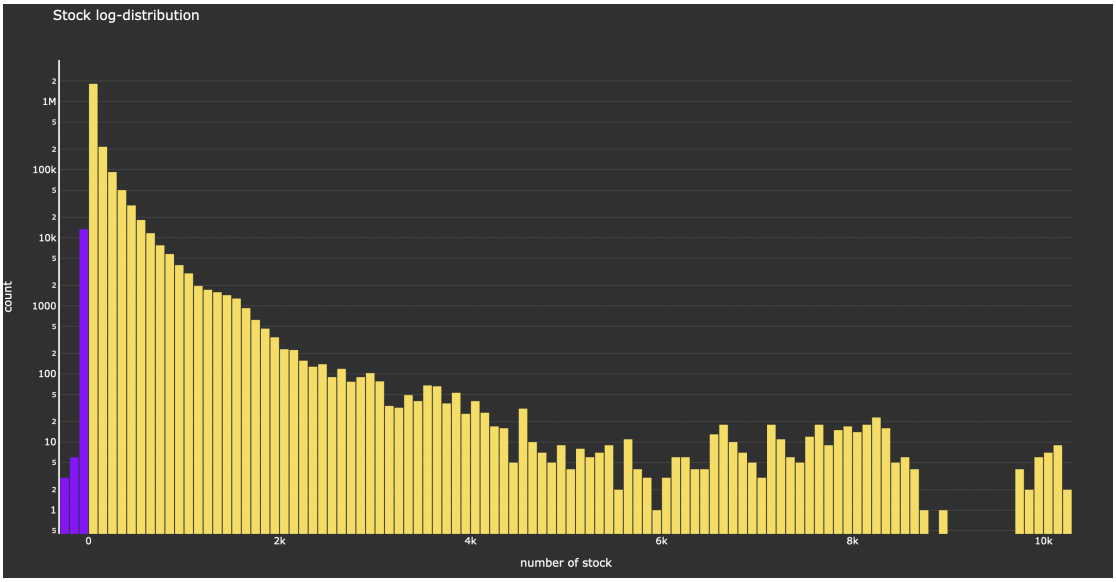
Prices_competitor contains all the prices for a specific product, for a specific seller and a specific date.
The dataset does not contain null values. The unique products_id are 6461 and there is no data on the remaining 1068 which are instead present in the *product_catalog*.

On average, for each day, there are records for 5393 unique products.

Also, there is a match between the number of sellers in the present dataset and the ones contained in the *sellers_list*.
It would be interesting to analyse the prices distribution for each seller and for each product over time in order to identify insightful patterns.

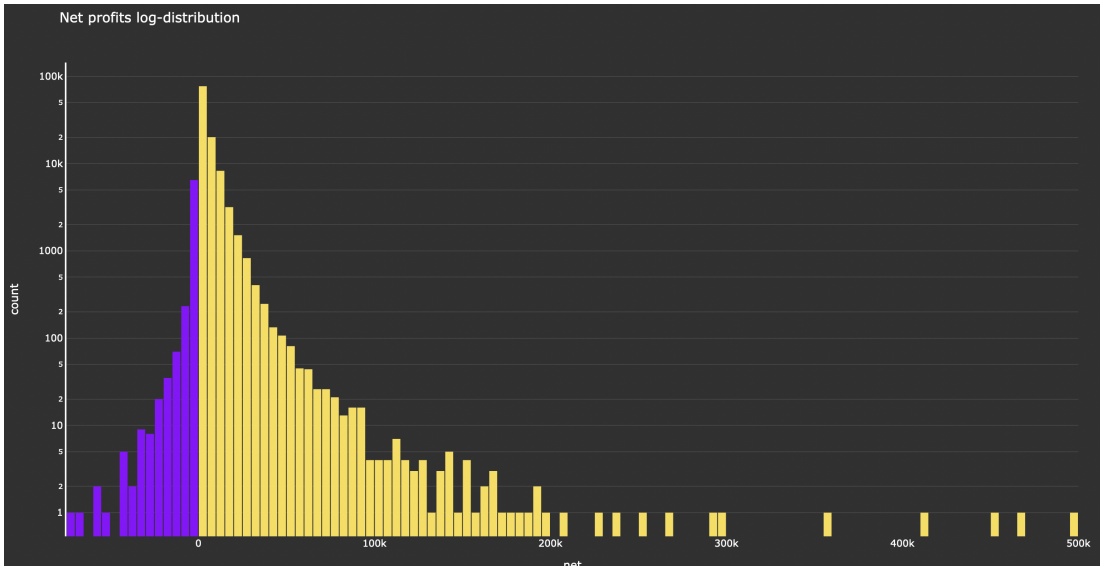
3 Stock

Stock contains the total stock for a specific product and for a specific date.
Stock values relate to a time interval that goes from 1 January 2021 to 31 December 2021.
Also in this dataset, there are no null values. There are stock data for all the products in the catalog.
On average, for each day, there are 6273 unique products whose stock value change.
The 75% of the total_stock is less or equal to 75 (3rd percentile), while the maximum stock value is 10247. The overall distribution of the total stock is right skewed, as can be seen in the plot below.
Additionally, the minimum value of the total_stock is negative, which is unexpected. This leads to a more in-depth analysis in relation to sales.
As a matter of fact, by joining *stock* with *sales_data* on date and product_id, it is possible to notice that there are 3818 cases in which a certain quantity of a product is sold even if, in the date of sale, the stock is less than that quantity.
However, it could be possible that in these cases the product is restocked in the coming days and it would not be a problem for the company to sell and deliver them on time.



4 Sales

Sales_data tracks all the purchases made by customers on the e-commerce website.
The dataset is complete, there are no missing values. Each row corresponds to a single transaction and provides information about the date of the purchases, the product id number, the number of products purchased, the sales price with and without taxation, the regular price, and the purchased price.
The time period runs from January 2 to December 31, 2021.
The product_id is a 6-digit unique identifier of the product and there are 7529 unique products that match with the value in the *product_catalog*.
The product quantity range for each transaction goes from 1 to 70.
The sales_price_tax is the final sales price for a single unit of the product which is affected by discounts and taxation (unlike the sales_price). Both the sales prices are at least equal to 1 up to 453303.
The regular_price_tax is the list price for a single unit affected by taxation, and it is at least equal to 1. Whereas the regular_price without tax has a minimum value of 0.833.
The regular price and the sales price are different if there are promotions or marketing campaigns for that product at that time. Hence, there is an inconsistency because the minimum list price is lower than the minimum price after a promotion is applied. The purchase_price is the cost at which a single unit of product is purchased.
After understanding the data provided, the taxation analysis was performed. The first step is to compute the difference between the price with taxation and without for both the sales and regular prices. Then, the difference between the taxes is computed and points out that the average of both taxes is 20%, with a maximum value of 20% and a standard deviation equal to 0.35. The focus is on the minimum value of taxation that is equal to 0. This data seems to be inconsistent since even primary goods should be taxed.
Later, analyses were computed at the business level. Revenues were calculated by multiplying between sales_price_tax and quanity for each transaction. Then, the profits were calculated as the difference between the revenues and the purchased_price.



5 Product catalog

Product_catalog contains all the information related to the features of a product.
The dataset does not contain missing values and shows that there are 7529 unique products, exactly as in the sales dataset.
The attribute product_id is considered a primary key. This implies that the dataset can be joined with all of the others containing product_id in order to add information about the products' categories.
In particular, each product belongs to 3 categories, which correspond to sub-categories of products on the site from the most general to the most detailed one. Instead of having category and brand names, there are codes. Products belonging to the same category 3 are substitute products.
The data are consistent with expectations because there are 12 unique categories in coded_cat1, 60 unique categories in coded_cat2, and 238 unique categories in coded_cat3. Furthermore, there are 292 brands.

6 Sellers list

Sellers_list contains the primary key for sellers, namely seller_id.
The dataset does not contain missing values and it is consistent because there are 9 unique sellers, each associated with a letter.
The e-commerce has as seller_id the number 24.

7 Clicks

Some data are available from a price comparator website specific for consumer electronics. The customer selects the product that wants to buy and the site shows different sellers of that same product with respect to 2 different sections:

- the bidding section in which there are the 3 sellers who paid to be at the top of the page,
- the regular section in which there are other 10 sellers in decreasing price order.

Clicks_regular and *clicks_bidding* have the same columns but a different time span. Bidding contains observations from 02-04-2021 until 04-01-2022 while *clicks_regular* contains observations from 01-01-2020 until 04-01-2022. These 2 datasets show the position of products shown in the comparator for each date. A concatenation between them allows creating a chronological record of all clicks. Some data cleaning and transformation must be performed before merging.

7.1 Clicks Regular

Clicks_regular contains observations regarding 6457 distinct product_id entries from 9 distinct sellers.

The dataset is the only one with missing values.
Moreover, there are 545 observations in which the position is 0.0 and 124 of these observation have also price, price_min and price_max equal to 0. Being aware that the comparator shows the positions in regular from 1 onwards and that 545 is 0.0004% of the dataset, it may be that these observations will not be considered by the next step of the analysis. This anomaly seems to be unrelated to the seller, date and product, so the observations in *clicks_regular* where position $\bar{0}.0$ will be removed, as they are considered entirely stochastic.
The position contains 51 distinct values from which 2 must be subtracted, which are position 0.0 and the position of null values. An interesting insight is that the positions clicked range from 1 to 32, the 104-106-109 and then move on to positions 1010 and later.
The position contains 1093664 null values for 8 out of 9 seller_id's, seller_id 490 has no null positions. The position null values seem disconnected from seller_id, to date and product_id.
The price, instead, has 1093244 null values for 8 out of 9 seller_ids, seller_id 490 has no null prices. As for the position variable, the null values of price appear to be disconnected from the seller_id, to the date, and to the product_id. When price is null, the position is also null but the opposite relationship does not hold (i.e. there could be null positions but with price).
Since the data sources obtained from the comparator are: *prices_competitors*, *clicks_regular*, and *clicks_bidding*, some missing prices are recovered through the price_competitors dataset. In particular, 87000 prices out of 1093244, about 8%, are recovered.
In order to check that the retrieved prices are accurate,the prices column of the *prices_competitor* are merged to the *clicks_regular* using the date, product, and seller as an index. The check is applied only to rows that do not contain null values. The result of this check reports that 90% of the prices have difference equal to 0. As a consequence, deriving the missing prices in *clicks_regular* by taking them from *prices_competitor* is a valid method.

7.2 Clicks Bidding

Clicks_bidding shows up sellers who paid to be at the top 3 position of the page. The position column accordingly contains values ranging from 1 to 3.
The dataset has no null values in any observation. The dataset contains 696999 observations, concerning 5797 distinct product_id inserted by 9 distinct sellers.