Farah Jabri

Alexandre Sukeratha

Lauryne Moyse

Dimitri Mades

MODS 203: Data analysis in economics 1

Final Report

SOMMAIRE

1 – Introduction

2 - Data collection

1. Background
2. Scrapping strategy
3. Overview of the features

3 - Data cleaning

4 - Data visualization

5 - Data analysis

1. Data description
2. Baseline - OLS approach
3. Case-by-case study of characteristics - development of new hypotheses
4. Change of perspective - ML approach

6 - Conclusion

1. Introduction

The topic of understanding the factors that influence the popularity of Amazon products is important because it can help businesses and sellers understand what drives consumer purchasing behavior and make informed decisions about product development and marketing strategies. Our motivation was to understand the Amazon Search algorithm. Indeed, Amazon being one of the largest e-commerce sites in the world, an algorithm for ranking items significantly changes not only Amazon's revenues but also the customer experience on the website.

When studying the Amazon website and searching for a subject, we ask ourselves: what are the parameters that influence the popularity of an Amazon product? This question is interesting because it addresses a commonly asked question among businesses and sellers looking to improve their sales on the platform, and it has the potential to reveal valuable insights about consumer behavior. More precisely, this lead us to the following research question:

What are the parameters that intervene in the ranking of search results on Amazon ?

To answer this question, we collected data on a variety of products from the Amazon website. We focused on 3 household appliances: coffee grinders, kettles and coffee machines. The data includes information about product ratings, reviews, pricing, characteristics of the product, the number of times the item has been rated. Once we have collected this data, we performed various analyses to understand the criteria by which amazon orders the items on its site. We used a combination of statistical methods and machine learning techniques to analyze the data and identify patterns or relationships between the different parameters and product popularity.

2. Data Collection

2.1.Background

On this subject, we were able to perform some research on the e-commerce strategy of Amazon such as the algorithm-based and user-generated recommendations, the algorithmic pricing system or the model followed to establish best-selling rankings.

For example, we found an article on the set of calculation rules that decide search results on Amazon called Amazon A9 algorithm. It is explained, in a nutshell, the main rules by which the algorithm ranks search results. In fact, the A9 algorithm is a search system used by Amazon to provide relevant results to users. It uses advanced techniques such as automatic natural language processing, machine learning and data mining to understand the intent of the user's search query and match it to relevant products. It also takes into account factors such as user behavior, purchase history and search history to provide personalized results. In addition, the A9 algorithm also uses a machine learning model to rank search results based on relevance, popularity and sales history.

2.2. Scrapping strategy

We encountered several problems during the second assignment. First of all, some of us could not manage to scrap data from their computer because Amazon was blocking access, even with headers. We were then left with only two computers to work with. We began to scrap the data with Python using the libraries Requests and BeautifulSoup . But the process was very time-consuming and crashed regularly (we have sometimes left loops running all night without getting a clear success). So we made the decision to use another way to scrap all the data. With JavaScript and the help of nodeJS, we managed to acquire the information we needed.

We first tried to use the same method as in Python, i.e. make http requests with particular headers. This method is the simplest and NodeJS supports query errors very well, so it is simple to know in detail what is not working. We used the "node-fetch" module and we just copied the query that our computer makes when searching for a product on Amazon but Amazon is not fooled and this method was not successful.

The last solution was to simulate a browser with NodeJS. It can be achieved in Python but it is not at all adapted, unlike NodeJS which is already a web language. So we used the "puppeteer" module which simulates a Google Chrome browser. The browser is told what to do, where to click or where to write, and it executes it as a human being would. We use asynchronous functions because the durations of the actions on the web pages are indeterminate and we must make sure that an action is finished before starting another one (we used the markers "async", "await" and "promise"). In this way we were able to go to each page and retrieve the data we were interested in with basic JavaScript functions that we inject into the page (getElementById, querySelector).

The data was initially in JSON format which is much more convenient than CSV because it is easily readable, very well ordered and can take both dictionaries and arrays. Afterwards we flattened the JSON so that the conversion to CSV would be done correctly and facilitate the analysis work in Python.

In order to avoid bugs, we collected the data in groups of 300 by forcing the algorithm to make a pause between each round. The data collection took a few hours since we performed it simultaneously on three computers (one for each type of item). We finally obtained 5721 scrapped items.

2.3. Overview of the features

The given criteria taken into account by the A9 algorithm provided us with leads for our study and the data to scrap. Unfortunately, we were not able to scrap all the features we wanted. Particularly tags such as "amazon prime" or "amazon choice”, as well as the number and type of multimedia content provided per article which all plays an important role in their visibility. Here is all the information we obtained:

Name	Type	Description
price	float	Price of the product in $.
rating	int	Customer's rating out of 5.
timesRated	int	Number of times the object was rated.
title	string	Title of the product.
link	string	Link to the product page.
page	int	Mumber of the page on which the product appears.
position	int	Article position in its page regarding its visibility.
ratings	string	Percentages of the different ratings obtained.
Color	string	Color of the product.
Brand	string	Brand of the article.
Material	string	Material used for the product.
Style	string	Description of the style of the coffee grinder.
Item Weight	string	Weight of the product in pounds.
Item Dimensions LxWxH	string	Dimensions of the item.
Capacity	string	Capacity of the product.
Recommended Uses For Product	string	Advice for the customer when using the product.
Specific Uses For Product	string	Tags for the product.
Product Dimensions	string	Dimensions of the product in inches.
Manufacturer	string	Manufacturer of the product.
ASIN	string	Amazon Standard Identification Number.
Country of Origin	string	Country of origin of the product.
Item model number	string	Model number of the product.
Customer Reviews	string	Rating and times rated #redundant.
Best Sellers Rank	string	Best seller rank of the product in a few categories.
Date First Available	string	Date when the announce was first released on Amazon.
Wattage	string	Wattage of the product.
Package Dimensions	string	Package dimensions.
Department	string	Tag for the coffee grinder.
Is Discontinued By Manufacturer	boolean	Is Discontinued By Manufacturer.
Voltage	string	Required voltage for the product.
Domestic Shipping	string	Details regarding where the item can be shipped.
International Shipping	string	Details regarding where the item can be shipped out of the USA.
Special Feature	string	Description of potential special features.
Coffee Maker Type	string	Type of the coffee grinder (Espresso machine, French press).
Human Interface Input	string	type of interface provided when interacting with the grinder.
Batteries	string	Model of the batteries (if required).
Filter Type	string	Type of filter to use for the cofee grinder.
Included Components	string	Additional components sold with the grinder.
Model Name	string	Name of the grinder model.
Number of Items	int	Number of items sold per package.
Package Type	string	Type of packaging.
Language	string	XXXX ?
Unit Count	string	Unit Count XXXX ?
Material Care Instructions	string	Instructions on how to maintain the equipment.
Assembly Required	boolean	Assembly required?
Number of Pieces	int	Number of Pieces.
Batteries Required?	boolean	Batteries required?
Material Feature	string	XXXX ?
Is Dishwasher Safe	string	Possibility to put the product in the dishwasher?
Fabric Type	string	Details on product compositio.
Import Designation	string	Imported?
Warranty Description	string	Details about warranty.

figure 1. Descriptive table of collected characteristics.

3. Data Cleaning

The cleaning was more challenging than we had expected. Although the items had a fairly common standard format, there were many differences from one product to another, but also within a category. Our cleaning strategy was based on several axes.

One of our tasks was to change the columns that were not in appropriate formats. This concerned for example the numerical values which appeared as strings or the dates which we put in a more suitable type.

A challenging part of the work was also to harmonize the data within the same feature. Indeed, we scrapped a lot of technical characteristics, but the values were not necessarily expressed in the same unity for all the items. This required additional research work to convert everything into a known unit.

The final step of the cleaning was to remove the eventual duplicates. We also added a new feature ‘features_nb’ which indicates the number of fields filled in on the page of an article and that we thought could be relevant. We finally decided, on an arbitrary basis, to remove the columns where more than 80% of the entries were non values.

4. Data Visualization

A first step in visualizing our dataset was to identify the country of origin of our products. This feature can be relevant when dealing with popularity criteria. Indeed, the COO can often influence the consumer's decision by acting as a guarantee of good product quality.

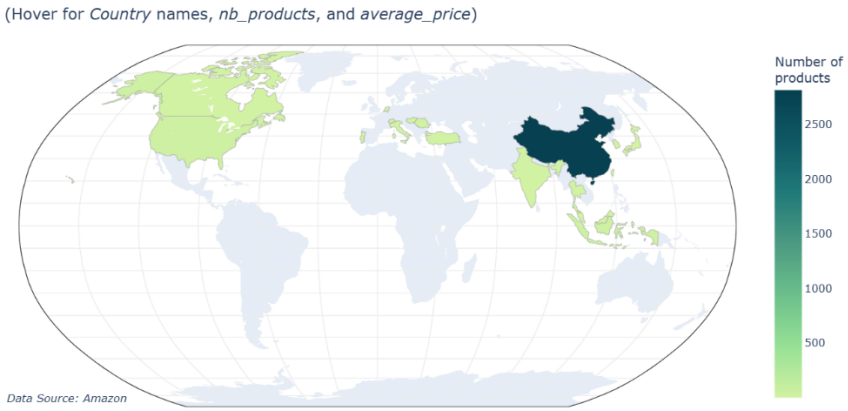

figure 2 – COO of scrapped Amazon products

Unsurprisingly, most of the product from the data we've scraped is from China.

The second step was to understand the link between the rate of a product and the number of times this product was rated.

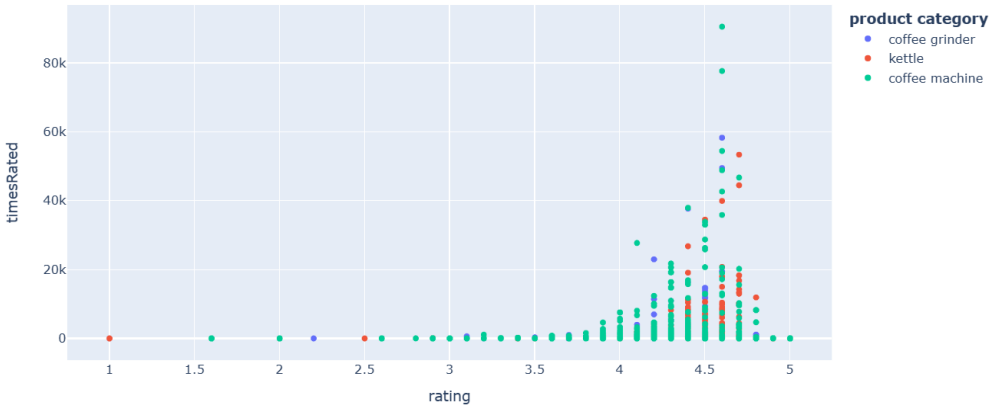

figure 3 – TimesRated as a function of rating

One thing that stands out in this figure is that highly commented products (rated time > 20k)all have ratings above four stars. Multiple hypothesis can be made from such results:

Snowball effect : a genuinely good product will get good reviews and new customers who will leave good reviews themselves.

Fake reviews : Products with a large number of reviews are more likely to be subject to false reviews that would increase their score.

Social influence : Large amounts of ratings can influence new reviews by dampening them.

cf. Sridhar, S., & Srinivasan, R. (2012). *Social Influence Effects in Online Product Ratings*. Journal of Marketing, 76(5), 70–88.

Then, we observed the price of the items we've scrapped, the vast majority of the products we handle are priced between $15 and $150 USD.

Finally, we've observed simultaneously the price, the ratings and the TimesRated.

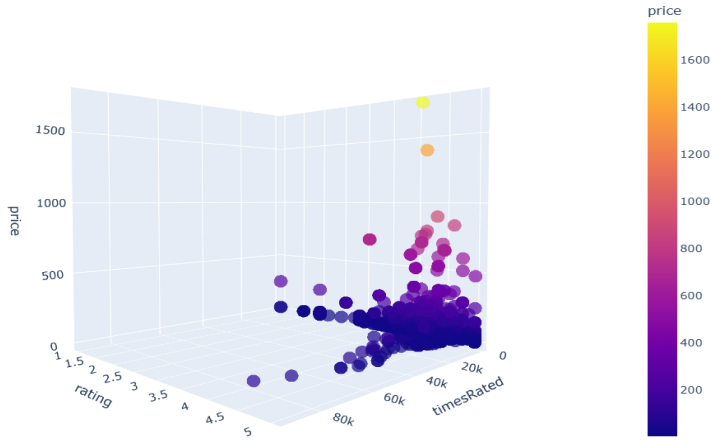

figure 4 – 3d Scatter plot between price, rating and timesRated

A large group of data points are clustered in the lower right corner of the cube. These are low price, high rating, many rated items. We can easily see outliers that differ from our identified main group : high price, high rating, many rated items / low price, low rating, not many rated items / low price, high rating, many rated items. In the end, all data points are collected on the edges of the cube.

5. Data Analysis

5.1. Data Description

	price	rating	timesRated	features_nb
count	5721.000000	5188.000000	5188.000000	5721.000000
mean	68.678762	4.292618	1800.510409	33.478413
std	91.256419	0.425137	4878.611437	6.216344
min	2.990000	1.000000	1.000000	19.000000
25%	24.990000	4.100000	25.000000	29.000000
50%	39.990000	4.300000	132.000000	33.000000
75%	79.990000	4.600000	1162.000000	38.000000
max	1757.980000	5.000000	90495.000000	49.000000

figure 5 – Descriptive statistics of 'price', 'rating', 'timesRated' and 'features_nb' features

From this table, we realized a quick analyze. We noticed that the products are globally really well rated (mean rating of 4.29 out of 5). There are also great disparities between products, the extreme values being very far from the average for certain features. It's the case of the price whose mean is 68.7$ while maximum value is 1758$. This kind of discrepancy is also found in 'timesRated' which indicates the number of times a product received a review. This can highlight the important differences in the number of sales of the different products.

5.2 Baseline : OLS approach

In developing our model for determining product positioning criteria on Amazon, we implemented an initial naive model whose performance will serve as a basis for further analysis. We performed a regression with an OLS method. The dependent variable therefore 'order', an int that indicates the order of appearance of an item after research. We considered its logarithm to perform the regression. The explicative variables are the first ones that came

to our mind (among the features we scrapped) and that we considered relevant for this model. It is the price, the rating, the number of times a product been rated (gives credibility to the rating) and the amount of information given by the manufacturer (also a guarantee of good manufacturing). Before running the regression, we looked at the correlation rate of the variables.

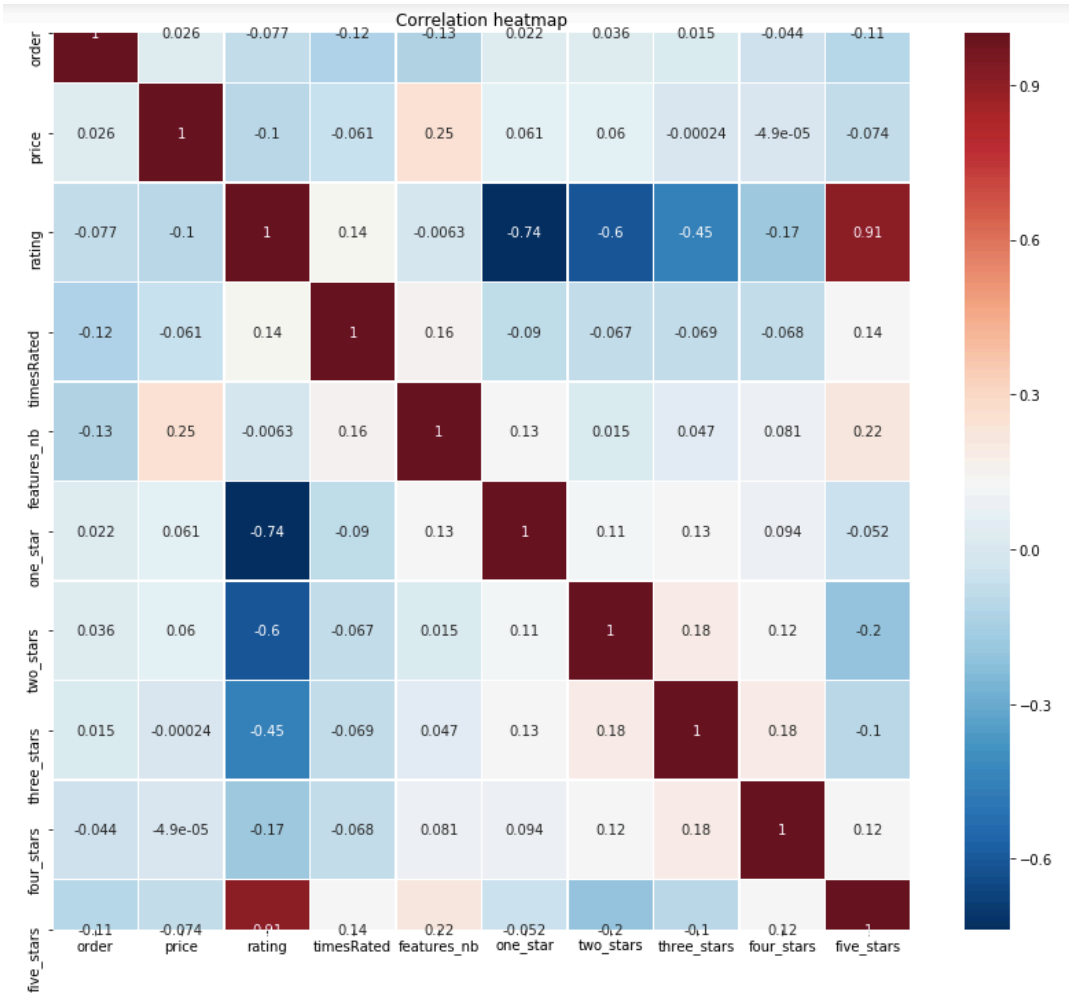

figure 6 – Correlation heat map between a few features

We can observe that the explanatory variables we chose are very poorly correlated, the coefficient varying from -0.0063 to 0.25. So there is no risk of having a multicollinearity problem. But we can also observe the absence of correlation between order and the other variables, the highest coefficient being 0.13.

We still performed the modeling that ran on 5188 observations. Here are the results :

	coef	std err	t	P>|t|	[0.025	0.975]
const	8.0817	0.160	50.424	0.000	7.768	8.396
x1	0.0003	0.000	1.941	0.052	-3.06e-06	0.001
x2	-0.1354	0.032	-4.169	0.000	-0.199	-0.072
x3	-4.63e-05	2.87e-06	-16.144	0.000	-5.19e-05	-4.07e-05
x4	-0.0262	0.002	-10.958	0.000	-0.031	-0.022

figure 7 – Summary of the OLS regression of 'order' by 'price', 'rating', 'timesRated' and 'features_nb' features

With x1 = 'price', x2 = 'rating', 'x3 = timesRated' and x4 = 'features_nb'.

The adjusted coefficient of correlation is equal to 0.088 which means that this model explains only 8.8% of the results. However, by taking a look at the distribution of the residuals, we noticed that it was close to a centered normal distribution., the estimator is therefore unbiased. Moreover, the p-value of the coefficients being small or even zero, we studied their value, more precisely their sign. The coefficient in front of `price` is positive. This means that the price increases with the order of an item. The cheapest items in this category are then displayed first. This observation appears to be coherent. The coefficients in front of `rating`, `timesRated` and `features_nb` are negative, which is also logical. The items displayed first have a higher rating, are rated more times and provide more information about the product.

Since the modeling was not really concluant in terms of fitting, we deciding to implement a forward variable selection to gain more information. We interpreted the order of appearance of the features in the algorithm as their relative importance in building the model. We considered the last explicative variables, and we added the five features that provides the percentage that a product obtained for each rating stars. Our results seemed to be quite coherent. The sub-features (percentage of opinions for each number of stars) were placed last. On the contrary, the primary features appeared first (`price`, `rating`, `timesRated`). We can conclude about the importance of the feature `timesRated` by trying to predict the feature `order` which was placed first on the appearance list).

5.3. Case-by-case study of characteristics - development of new hypotheses

Our initial econometric approach failed to find useful clues to explain the visibility of a product based on its technical and economic characteristics. In this section, we will use a case-by-case approach to find models that could help us better understand the underlying algorithms used on the Amazon website.

5.3.1 Influence of the ranking in the visibility of a product

An initial visualization of Data did not indicate a clear positive correlation between a product's ranking and its visibility. In fact, you have to think in terms of sales. Highly rated products are not necessarily the best-selling products. A highly rated product with a high price (luxury products) has a smaller audience than an average cheap item.

Nevertheless, products with an average rating of 3 or less only appear after the 14th page. One thing that could be interesting for future analysis would be to implement a new boolean function ‘rating_less_than_3_stars’ which would surely have a great correlation with the ranking function.

5.3.2 Between practicality and aesthetics – materials used

We wanted to know if the material used (plastic, metal, etc.) has an actual influence on the popularity of the product. We first use a naive visual approach to identify patterns linking the material used to a potential increase or decrease in score. This first approach did not give any clear results.

A second approach was to consider the number of comments instead of the score. To do this, we plotted for four key elements (presence of plastic, metal, stainless steel and glass) the average evaluation times of related products.

We noted that products containing glass received more reviews on average. We do not believe that this result is really significant. But it can be interpreted as the fact that glass kettles are preferred to metal or stainless steel kettles for aesthetic reasons.

5.3.3 Impact of colors on scales

As the rating feature did not give good indications when it comes to aesthetic features. We will continue to use the number of comments as a feature of interest. We displayed a

visualisation of the average evaluation time of a product according to the colours in which it is sold. Our first reaction reading the graph : White, metallic and gold colours are the most popular items. This is not surprising, as a device is often purchased in the most subdued colour possible.

In fact we have to be careful about the conclusions we can draw. Indeed, a colour that is too generic and often easy to apply (metallic often corresponds to raw material) offers results that are difficult to interpret. Moreover, these same colours are proposed by all manufacturers. Thus, gathering all the ranges of products: good note, great number of sales or items little known and badly noted.

4. Change of perspective, ML approach

The results obtained with the data analysis gave us clues on what influences the popularity of a product but as the results were very heterogeneous, we decided to use a machine learning approach to check whether we could really predict it or not.

We cleaned the data to keep only the most common fields for all products: keeping the voltage or the Amazon reference of the object in question is not necessary. With this simplified database, we used a Decision Tree Classifier (using pandas and sklearn) which allowed us to obtain a certain result. The algorithm for predicting featured items on Amazon only gave a prediction of 20%, which isn't satisfying at all. Yet, two conclusions could be drawn. First we could question our dataset : was it large enough to predict with a high level accuracy ? Were there too many NaN values ?

On the other hand, we can question the algorithm of Amazon itself. Amazon may put in the forefront the products it subcontracts or the products it manufactures. Since we haven't the data that says whether an item was manufactured or subcontracted by Amazon, we weren't able to visualize or predict these kinds of conclusions. In addition, it is possible that Amazon's algorithm uses user cookies, therefore adapting to each user.

6. Conclusion

The algorithm for predicting featured items on Amazon only gave a prediction of 20%, this probably means that it is not able to accurately predict which items will be featured on Amazon just by looking at their features or their rating and numbers of ratings. This also confirms our analysis of the data: we found it very difficult to determine which features affect the popularity of a product. Amazon's ranking is therefore not perfectly explainable by objective variables, which are what we focused on. It is maybe necessary to take into account Amazon's business strategy and goals. The company may choose to highlight certain products to drive sales or to promote certain brands or categories. For example, Amazon may feature products that are part of exclusive partnerships or advertising agreements with other companies. It may also promote items that are part of its own private label or exclusive brands. With more time to realize this project, we believe we would have scraped more relevant features (see II-3) whose absence is undoubtedly one of the biggest shortcomings in our project. It would have also been interesting to compare the search results of the different Amazon platforms (e.g. Amazon.com Com VS Amazon.fr)