TÉLÉCOM PARIS

MODS207- Project in Applied Economics

Ulrich Leitenberger

How User Information Drives Amazon’s Products Recommendations

Daniel Jorge Deutsch
Kevin Felipe Kühl Oliveira

Paris, June 1th 2021

Summary

Figures

Tables

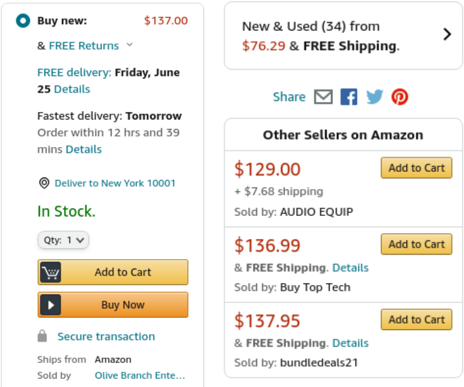
1. Introduction

It is a well-known fact that many online platforms personalize their content based on user information. This customization can be either advantageous or disadvantageous to the user.

In many cases, personalization improves the overall user experience (UX): Netflix uses previously watched movies to recommend other movies that the user might be interested in watching. Social media applications such as Facebook, Twitter and Instagram use the so-called “bubble filter” to only show users the contents they might be interested in. The same principle goes for platforms such as Youtube, LinkedIn, etc.

In other cases, however, personalization can be used to take advantage of the user. E-commerce platforms such as Amazon, Walmart, etc. are believed to use user information to perform price steering and price discrimination. In other words, these websites make use of user’s information to manipulate what products are displayed (and in which order) for a given search query and also what pricing strategy will be implemented for these products. Both of this behaviours can be hard to detect, in special the price discrimination, as e-commerces try their best to hide their discrimination and steering techniques.

Another relevant topic is the eventual impact of personal information on the suggested sellers (known as the buy-box sector of a given product). What are the factors that define the way platforms suggest sellers for a given product? Are those sellers' lists always the same for all kinds of users? Those are interesting questions that aim at unveiling an unclear pricing strategy.



○ Buy new: $137.00
& FREE Returns
FREE delivery: Friday, June 25 Details
Fastest delivery: Tomorrow
Order within 12 hrs and 39 mins Details
Deliver to New York 10001
In Stock.
Qty: 1
Add to Cart
Buy Now
Secure transaction
Ships from Amazon
Sold by Olive Branch Ente...

New & Used (34) from $76.29 & FREE Shipping.
Share

Other Sellers on Amazon

$129.00 Add to Cart
+ $7.68 shipping
Sold by: AUDIO EQUIP

$136.99 Add to Cart
& FREE Shipping. Details
Sold by: Buy Top Tech

$137.95 Add to Cart
& FREE Shipping. Details
Sold by: bundledeals21

Figure 1 Amazon’s Buy-Box

In this project we aim at exploring these three phenomena related to the use of personal data in ecommerce results and pricing personalisation.

2. Resume

The proposed analysis will take into account data collected from Amazon as it is considered to be the most relevant platform in the e-commerce universe. The first part of the analysis consists of a brief descriptive analysis followed by examination of evidence of price steering, price descrimination and finally the analysis of influential factors in buy box suggestions.

The analysis of price steering mechanisms made use of metrics such as the Jaccard Similarity Index and Kendall's τ. Price descrimination strategy was evaluated through the analysis Kendall's τ. Finally, buy-box suggestions were explored using Spearman's Rank Correlation and a random forest model.

At the end, strong signs of price steering were found. Evidence of price descrimination, however, wasn't significant. Finally, price proved itself the most impactful factor in the suggestion of sellers on the buy-box (not always in benefit of the user).

3. Empirical Strategy

This study is mainly divided into two types of analysis: personalization and buy-box recommendation. In the first one, our goal is to verify if Amazon performs price steering and/or price discrimination based on user's information. In the second one, we intend to analyse which of the seller's features drives Amazon's choice of buy-box for each product.

To properly establish our approach on simulating different users in the browser, we had to firstly understand what information Amazon could use to personalize its search results. After reading the article *Measuring Price discrimination and Steering on E-commerce Web Sites [1]* it became very clear that Amazon could use any of the information it receives on the HTTP requests sent to their servers to personalize their search results. Therefore, data such as IP addresses, operating system, browser, browser history, cookies, etc. should all be carefully set before every single request sent to Amazon's server.

3.1 Definition of the Personas

To find evidence of customized search results we should, of course, simulate different personas accessing the same information. To do that we have to play with the aforementioned data that Amazon receives from the HTTP request and, with those parameters, try to induce Amazon into thinking we have solid interest in certain activities/objects. Since there are so many variables to consider, we decided to focus solely on the effects of cookies and browser history on customization. Therefore, all the simulated personas would be using the same browser and operating system to enable comparisons over the *Ceteris Paribus* condition. It is important to note that even though we didn't use the same IP address for each persona (to not influence the results), all of the IPs were located in the same city.

Parameter	Value
Operating System	Windows 10
Browser	Chrome 91
IP Location	New York

Table 1 Static Personas Parameters

Once the simulated parameters were established, we could, then, define our personas:

Persona Names							
sports	shopping	science	home	health	games	dummy	computers

Table 2 Simulated Personas

Notice that, in addition to these seven personas, we've also added a "control" persona - named "dummy" - without any cookie nor history, to facilitate unbiased comparisons.

Having established all the personas, we needed to properly simulate them in the browser, i.e., we had to find the right urls to add to our history and the right cookies to add to the browser session. To do so, we turned to the website similarweb which listed the top 100 most accessed websites of each field. For each one of the personas we, then, added the suggested urls to the browser session history and accepted their cookies before going to the Amazon website.

3.2 Definition of the Search Queries

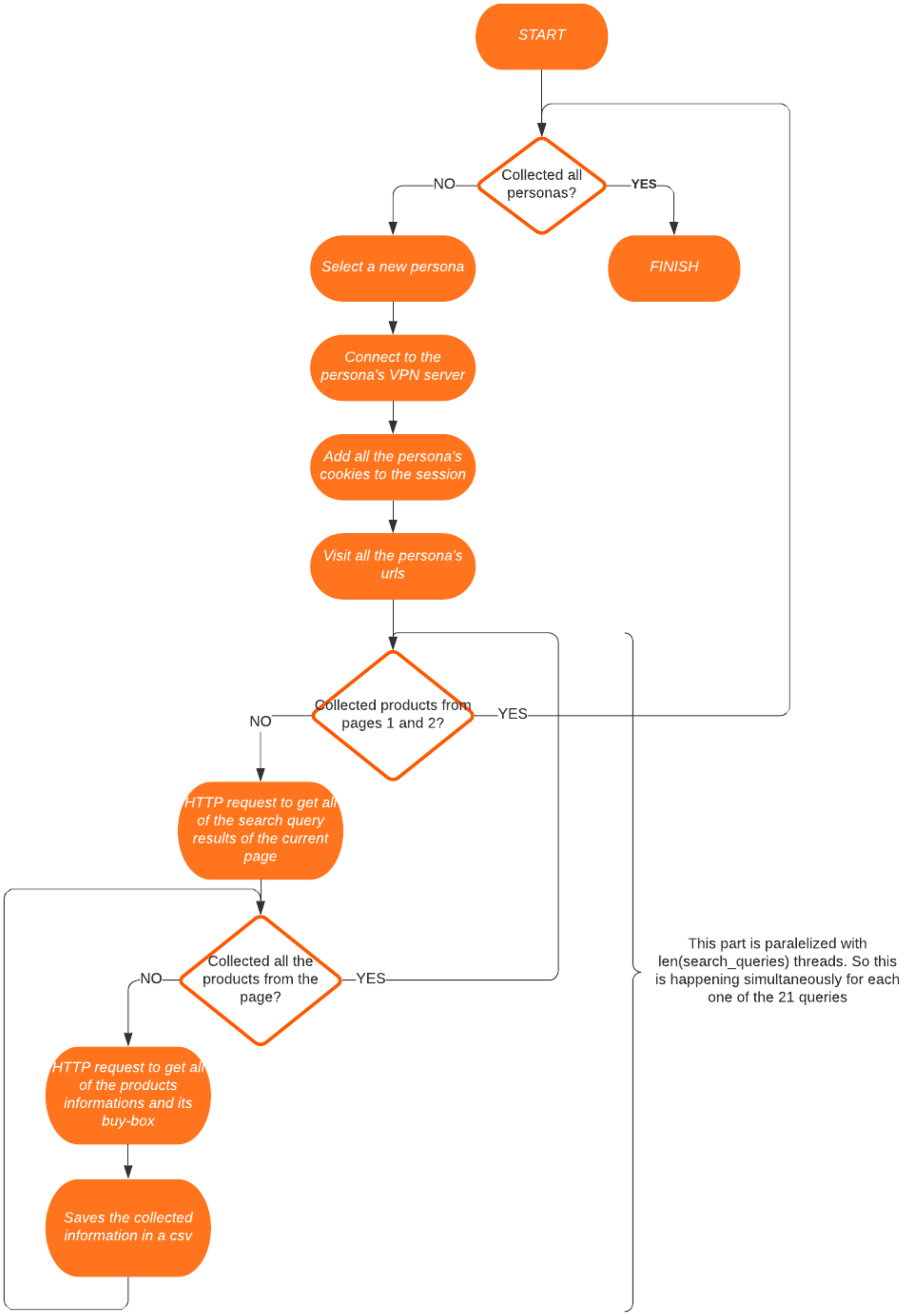
Once each persona was defined, we decided to create three search queries per persona - 21 search queries overall - to avoid coincidences in the results. The goal here was to think of queries that the corresponding persona would be very likely to do on Amazon, but the other personas not so much. This way, we decided to explore the following:

Personas	**Search Queries**		
Computers	Macbook Pro	Dell Xps	Webcam
Games	Playstation	Joystick	Gaming Headset
Health	Body Weight Scale	Omega 3	Body Lotion
Home	Carpet	Lamp	Silverware
Science	Telescope	Weather Station	Arduino
Shopping	Sunglasses	T-shirt	Purse
Sports	Baseball Bat	Basketball	Punching Bag
Dummy	-	-	-

Table 3 Search Queries per Persona

4. Data Collection

For each one of the personas we used the following flow to collect the data:


START
Collected all personas?
NO
YES
Select a new persona
FINISH
Connect to the persona's VPN server
Add all the persona's cookies to the session
Visit all the persona's urls
Collected products from pages 1 and 2?
NO
YES
HTTP request to get all of the search query results of the current page
Collected all the products from the page?
NO
YES
HTTP request to get all of the products informations and its buy-box
Saves the collected information in a csv

This part is paralelized with len(search_queries) threads. So this is happening simultaneously for each one of the 21 queries

Figure 2 Collection Flow

It is worth mentioning that Amazon is very strict in its anti-scraping policies and, because of that, they are very good in detecting automated access to their website and very quickly blocks the IP address that sent the request. To avoid that, we implemented a function that makes use of a *backoff_factor* that, every time we got a blocking response from amazon, we would wait *backoff_factor*$*2^i$ seconds (where i is the number of times we retried the same request) until retry sending the request.

The data collection process took about 4 hours. This time, while relatively low for such a voluminous scraping process on a site like Amazon, is a bit longer than ideal. This is due to the fact that Amazon updates its product prices constantly, and since each person's search results were collected sequentially (rather than in parallel), this may present inconsistencies in the data that imply price discrimination or price steering, even if it is not.

In an ideal collection scenario, we would have at our disposal a dedicated machine to collect search results for each persona, so that the process would be parallelized for both personas and queries (and not only for queries as it turned out to be our case).

5. Data Description

Once the collecting process is finished we can analyse what we managed to gather. Firstly, let’s take a look at the structure of the information we gathered. Each row of the collected data has attributes that could be divided into 4 groups: search, persona, product and seller.

Attribute	Example	Description
date	“2021-06-14 09:13:29.630244”	Date when the collection occurred
search_query	“gaming headset”	Query searched on the website
url_bb	“https://www.amazon.com/gp/aod/ajax/ref=auto_load_aod?asin=B00SAYCVTQ”	Url used to collect buy-box information

Table 4 Search-Related Attributes of the Collected Data

Attribute	Example	Description
persona_name	“dummy”	Name of the persona impersonated in the request
vpn_country	“United States”	Country where the VPN server was located
vpn_city	“New York”	City where the VPN server was located
vpn_server_ip	“62.182.99.93”	IP of the VPN server
user_agent	“Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.101 Safari/537.36”	User-Agent header param used in the HTTP request

Table 5 Persona-Related Attributes of the Collected Data

Attribute	Example	Description
prod_asin	“B00SAYCVTQ”	Amazon Standard Identification Number
prod_name	“HyperX Cloud II - Gaming Headset, 7.1 Surround Sound, Memory Foam Ear Pads, Durable Aluminum Frame, Detachable Microphone, Works with PC, PS4, Xbox One - Gun Metal”	Name of the product
prod_condition	New	Whether the product is new or used
page	1	Page where the product was
prod_index	14	The position on the page where the product was
prod_n_reviews	6606	Number of reviews of the product
prod_rating	4.6	Average rating of the product
prod_price	79	Price of the product
prod_shipping	None	Price of the product shipping

Table 6 Product-Related Attributes of the Collected Data

Attribute	Example	Description
seller_name	Amazon.com	Name of the seller
seller_index	0	Position of the offer on the buy-box
seller_rating	94	Rating of the seller
seller_n_reviews	27285	Number of seller reviews

Table 7 Seller-Related Attributes of the Collected Data

In the figure below we can see how many products and product offers were collected for each persona. Overall we see that with a 1.22% margin, all the personas were shown the same amount of products and, but the amount of offers differ a bit more (4%), which could be evidence of price steering.

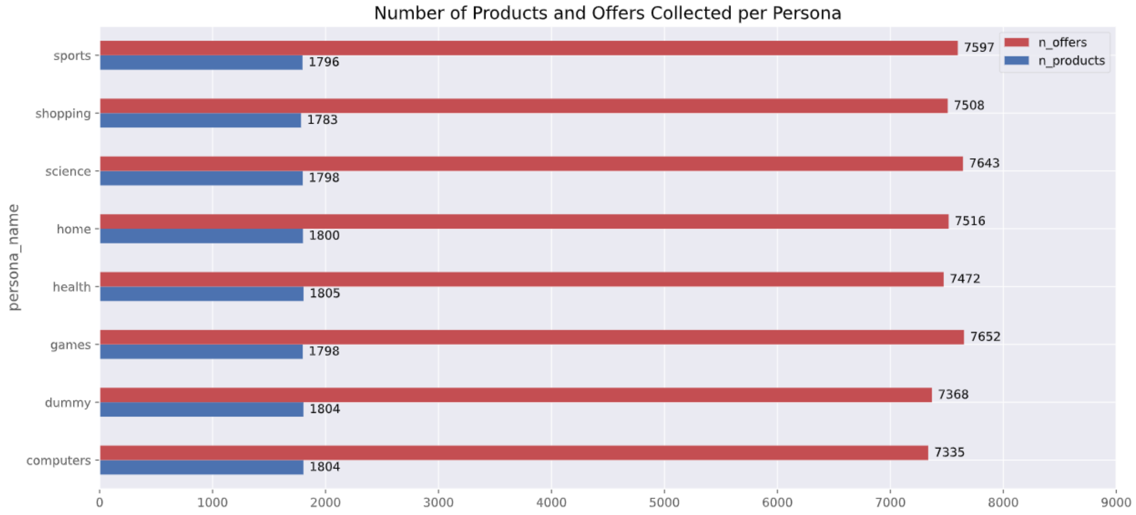


Figure 3 Number of Products and Product Offers per Persona

Now, if we observe the number of products and product offers per search query (figure below), we can clearly see that some products tend to have more sellers on the buy-box than others.

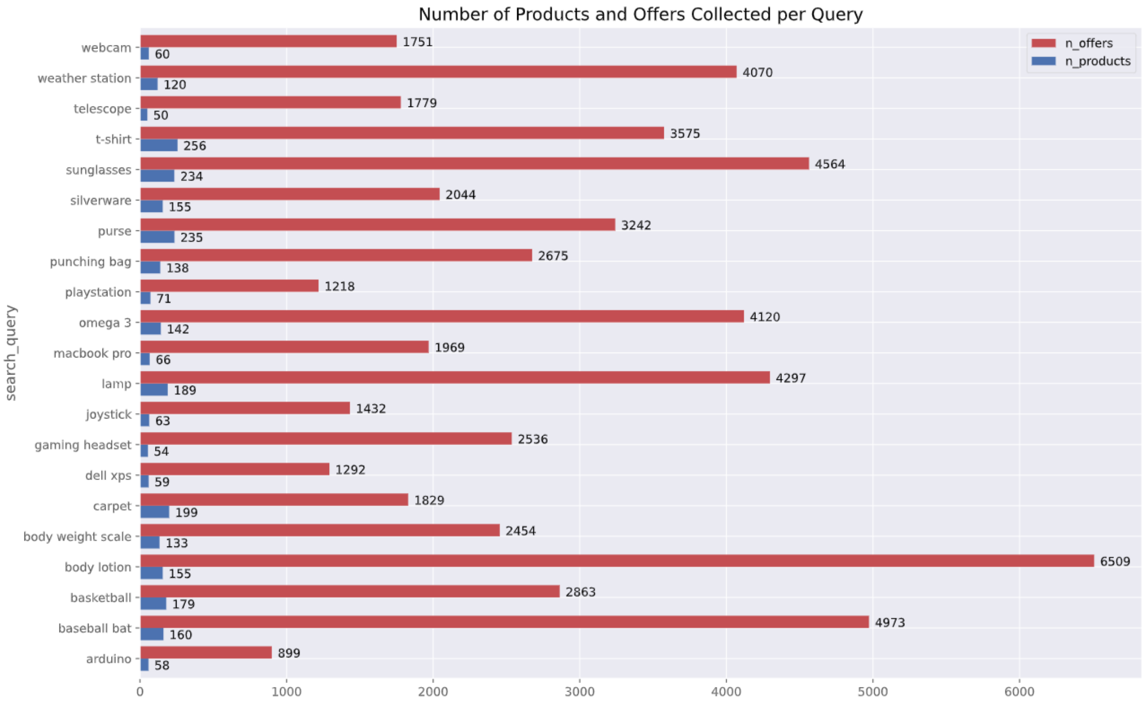


Figure 4 Number of Products and Offers Collected per Query

The plot above also suggests that the quantity of products displayed on the page depends on the search query. On the plot below we can clearly see that that's the case, and moreover, we can actually observe that electronics and more expensive products tend to have fewer items per page.

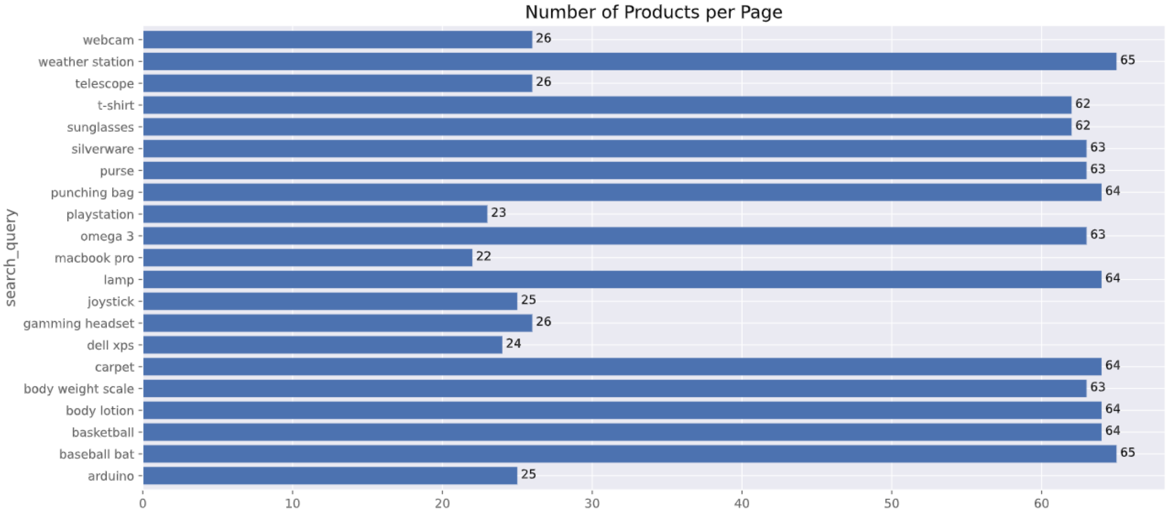


Figure 5 Number of Products per Page

6. Analysis

In the previous section we were already able to observe some differences between the obtained search results for each persona. In the following figures, we are able to see the difference between the average price of the search results for each search query and persona. The information shown on these plots could mean that Amazon price steers and/or price discriminates based on user information.

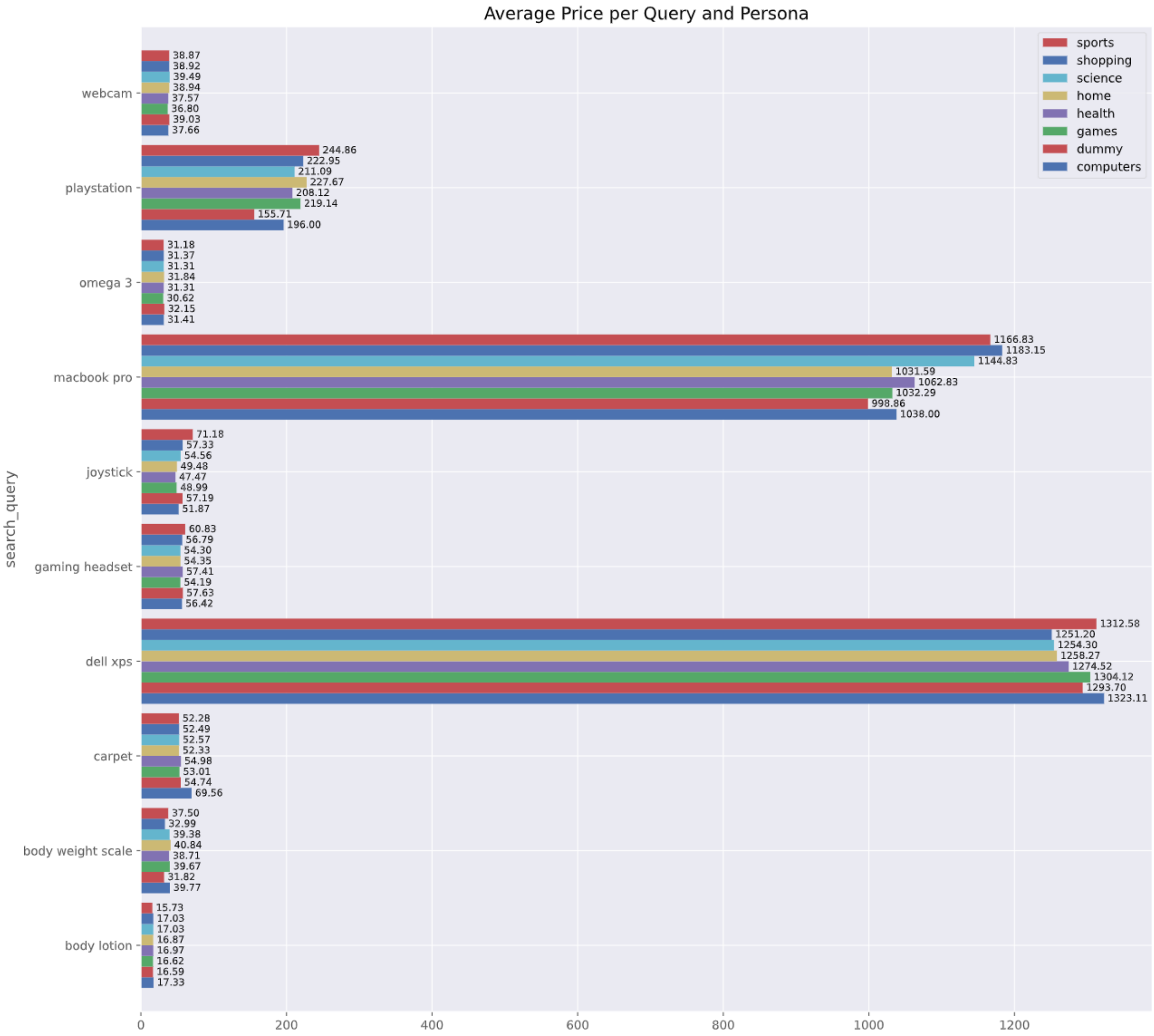


Figure 6 Average price per Query and Persona (First Half)

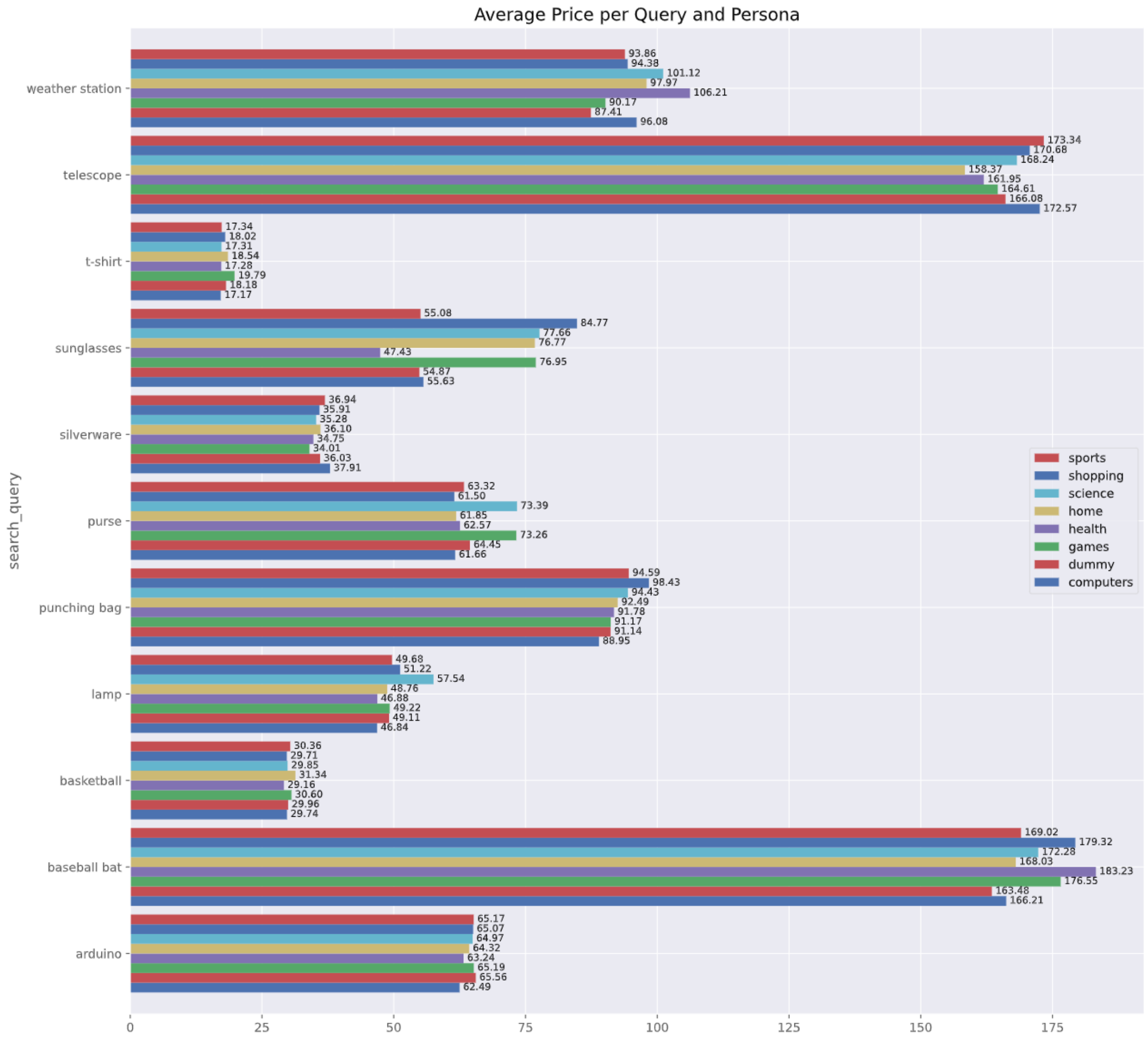


Figure 7 Average price per Query and Persona (Second Half)

If we take a detailed look on the above results, we can see that some search queries have more variation than others. We notice that "dell xps" is more likely to show more expensive results for people more interested in computers, games and shopping. A sunglass tends to be more expensive for those who are more into shopping than for those into computers.

It is important to also notice that not all the data presents the results we expected. A purse, for instance, was more expensive for those into science and games than for those into shopping. We believe that these results were obtained either because of one of three reasons: some sort of intersection in the interest of the personas (for example, a lamp could be interesting for each one of the personas, but with different goals in mind), misleading - or too general - information on the cookies (for instance, the websites visited for the "health" persona were much more related to mental health than physical health) or because of the duration of the collection process (it took 4 hours in total, which gives Amazon enough time for some small price variations).

6.1 Price Steering

As defined before, price steering is basically when two users receive different product results (or the same products in a different order) for the same query. Figure 3 actually shows us some evidence of price steering in our data because every product has a fixed number of sellers and thus offers. Therefore, the only reason why each persona obtained a different number of offers (with a relatively high margin) is because it presented different products to them.

6.1.1 Jaccard Similarity Index

The Jaccard Similarity Index is a very simple metric that measures the overlap between two different sets of results. It is given by the following formula:

$$J_{(A,\,B)} = \frac{|A \cap B|}{|A \cup B|}$$

A Jaccard Similarity Index of 0 represents no overlap between the results, while 1 indicates they contain the same results. To calculate the index without any bias, we compared all the personas with the “dummy” (the persona collected without any cookies). Therefore, the closer the index is to 1, the more similar the results for this persona when compared to the “dummy” persona. On the other hand, the closer the index is to 0, the more distinct the displayed products for the persona are from the “dummy”.

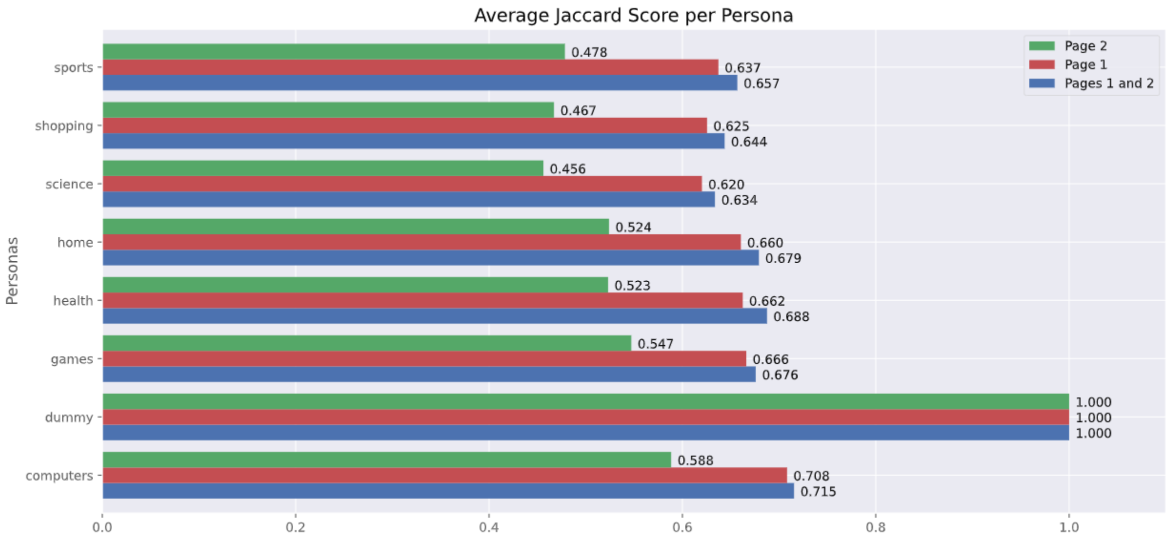


Figure 8 Average Jaccard Score per Persona

From the plot above, it is clear that Amazon performs price steering based on user information since neither of the results gets close to 1 (except for the “dummy”, because it is being compared with itself). We can also notice that the results start to diverge even more when you consider the second page of results.

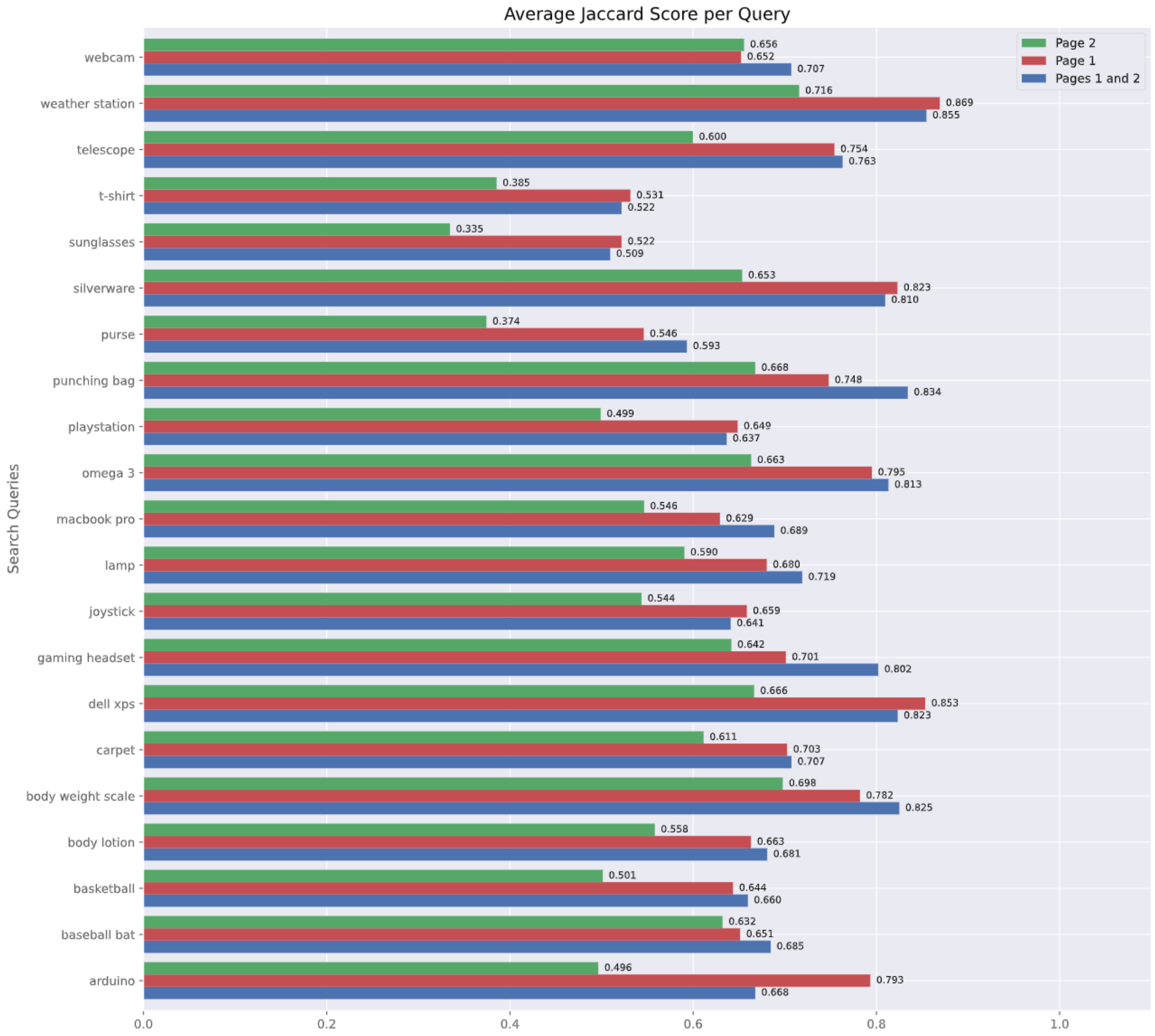


Figure 9 Average Jaccard Score per Query

In the plot above, we can even examine which of the search queries produces the most divergent results based on user information. Note here that queries for more specific products like "weather station", "dell xps" and "omega 3" tend to have less evidence of price steering than more generic queries like "t-shirt", "purse" and "sunglasses".

6.1.2 Kendall's τ

Even though the Jaccard Similarity Index already gave us proof that Amazon price-steers based on user information, price-steering involves more than just the intersection of displayed products. As previously stated, price steering also happens when the identical products are displayed in a different order, and the Jaccard Index is unable to detect this.

To measure this variant of price steering, we used the Kendall's τ Coefficient, which is given by the following formula:

$$\tau = \frac{P-Q}{\sqrt{(P+Q+T)(P+Q+U)}}$$

Where P is the number of concordant pairs, Q the number of discordant pairs, T the number of ties only in x, and U the number of ties only in y. Thus, this coefficient varies in the interval [-1, 1] , with 1 representing the same order of search results, 0 signifying no correlation and -1 being inverse ordering of search results. Here, again, to calculate the index without any bias, we compared all the personas with the "dummy" (the persona collected without any cookies).

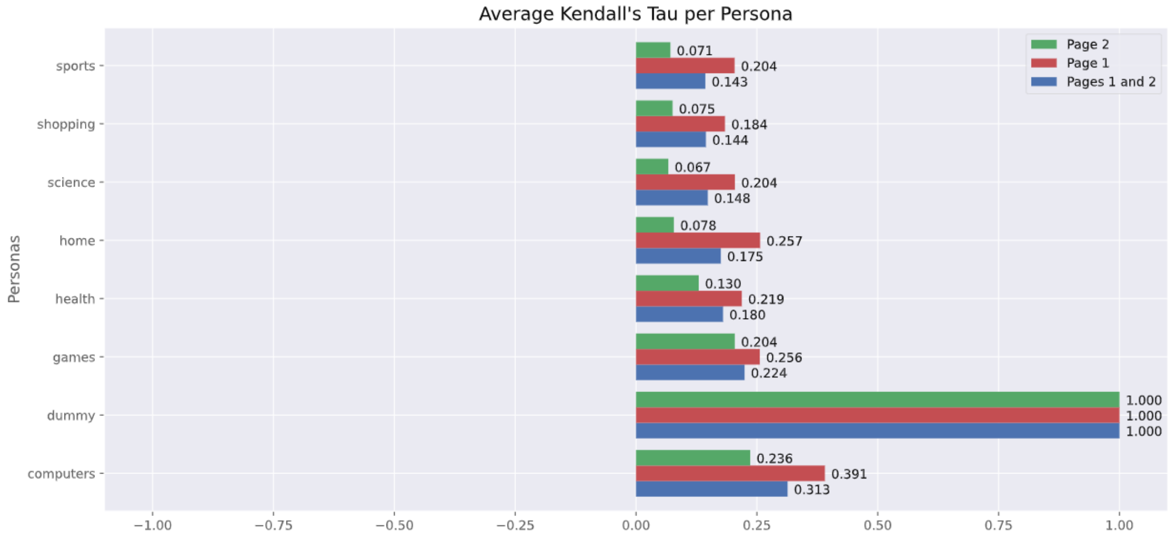


Figure 10 Average Kendall's τ per Persona (Price Steering)

From the plot above, it's clear that Amazon changes the order of the search results based on the user's personal information as neither of the results is close to 1 (except for the "dummy", because it is being compared with itself). We can notice, though, that Amazon doesn't go as far as reversing the list of search results as neither of the results are negative.

We also notice that Amazon tends to make more changes on the results of the second page rather than the first page.

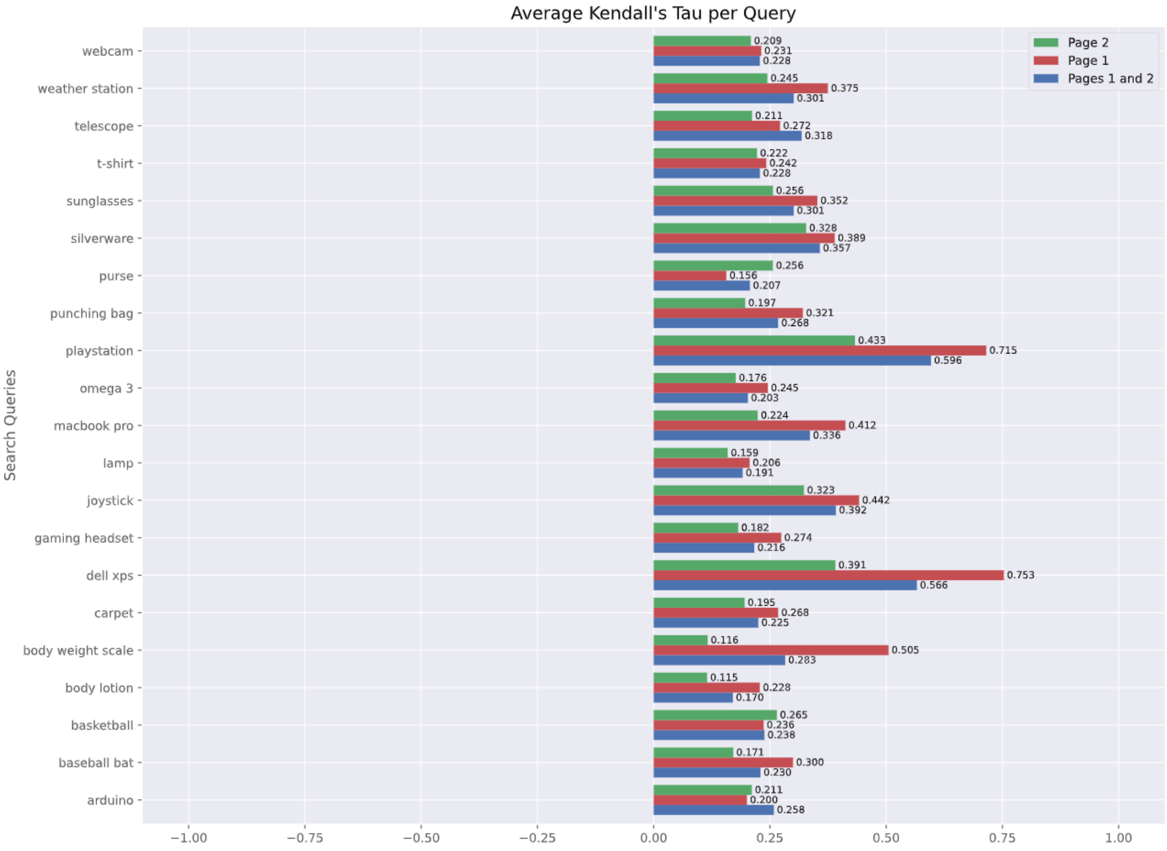


Figure 11 Average Kendall's τ per Query

In the plot above, we also examine which of the search queries produces the most divergent results - considering the disposal order - based on user information. Here, we notice that search queries such as “playstation”, “dell xps” and “body weight scale” tend to maintain more the order of results whereas queries like “purse”, “lamp” and “body lotion” tend to diverge more on the results for each persona.

6.1.3 Normalized Discounted Cumulative Gain

Now that we’ve encountered evidence of both “variants” of price steering, we would like to measure how strongly the ordering of search results is correlated with product prices. To do so, we use the Normalized Discounted Cumulative Gain (nDCG).

Each possible search result *r* is assigned a “gain” score *g(r)* - in our case, *g(r)* is the price of the product *r*. The DCG of a page of results *R* = [*r1*, *r2*, ..., *rn*] is then calculated as:

$$DCG(R) = g(r_1) + \sum_{i=2}^{n} \frac{g(r_i)}{log_2(i)}$$

The Normalized DCG is simply given by:

$$nDCG(R, R') = \frac{DCG(R)}{DCG(R')}$$

Where *R'* is the list with all the search results sorted from greatest *g(r)* to least. This way, the nDCG varies in the interval [0, 1]. An nDCG of 1 means the observed results are the same as the ideal (*R'*) results, i.e., the order of disposed products tends to coincide with the order of products sorted from most expensive to least. On the other hand, an nDCG equal to 0 means that no useful results were returned.

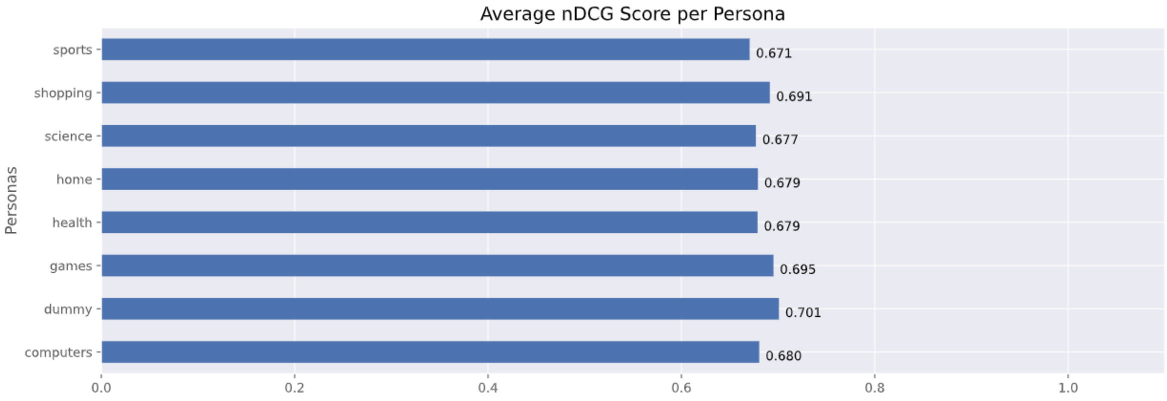


Figure 12 Average nDCG Score per Persona

From the plot above we can notice that the disposal of the search results is influenced by the price of each product, but not to the point of being a descending order of prices. We observe that there isn't much difference between the nDCGs of the personas but, besides that, we can see that the nDCG of the "dummy", is the highest one, which means that if Amazon doesn't have enough information about the user, it will tend to dispose higher priced products on the top of the search results. Following that thought, we see that "games" and "shopping" are the personalities with the highest nDCG, which could mean that Amazon sees these personalities as the ones with higher willingness to pay when compared to the others.

6.2 Price Discrimination

As previously stated, price discrimination is when two personas are charged different prices for the same product from the same seller. When we look at the figures 6 and 7 we might be inclined to believe that Amazon does some sort of price discrimination because we se a different average price for each persona on the same query. This belief, though, doesn't prove that Amazon price discriminates because the obtained results could have happened only because of price steering (which we proved that is performed by Amazon in the previous section).

6.2.1 Kendall's τ

To search for price discrimination we also make use of the Kendall's τ, but this time, instead of comparing the only the asin of the products obtained for the persona and for the "dummy", we compare the asin of the product and its price. This way, if τ is close to 1, it means that there is not much difference between the price charged for the persona and for the "dummy". Having a τ is close to 0 would mean that there is no correlation between the prices charged for the persona and the "dummy". A τ close to -1 would mean that prices are in the inverse ordering.

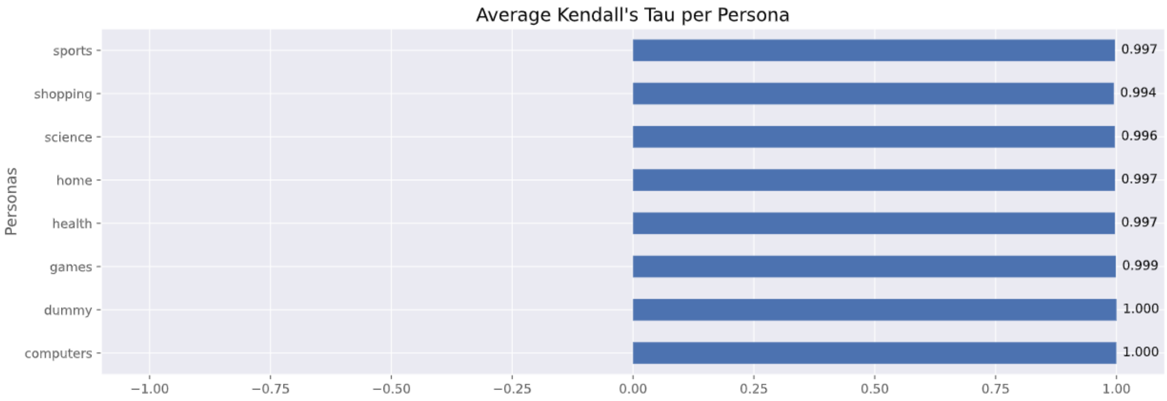


Figure 13 Average Kendall's τ per Persona (Price Discrimination)

From the results shown above we can see that we obtained no conclusive evidence of price discrimination based on user information in the Amazon platform. The small differences between the personas are probably due to small chances of product prices that Amazon does throughout the day and, since our collection process took about 4 hours to finish, some of the personas were affected by it.

6.3 Buy-Box

6.3.1 Influence of the Price

The first thing we wanted to understand from the buy-box was if Amazon would always pick the seller with the lowest price as their recommended seller. To verify that, for each persona, search query and product, we calculated the Spearman's Rank Correlation Coefficient between the product offers sorted by *prod_price* and the product offers sorted by *seller_index*. This coefficient is a measure that varies in the interval [-1, 1]. The closer the coefficient is to -1, in our context, means that Amazon's recommended seller for each product would always be the one with the highest selling price. On the contrary, the closer the coefficient is to 1, would mean that Amazon's recommended seller for each product would always be the one with the lowest selling price. Finally, if the coefficient is equal to 0, it would mean that there is no relationship between the variables.

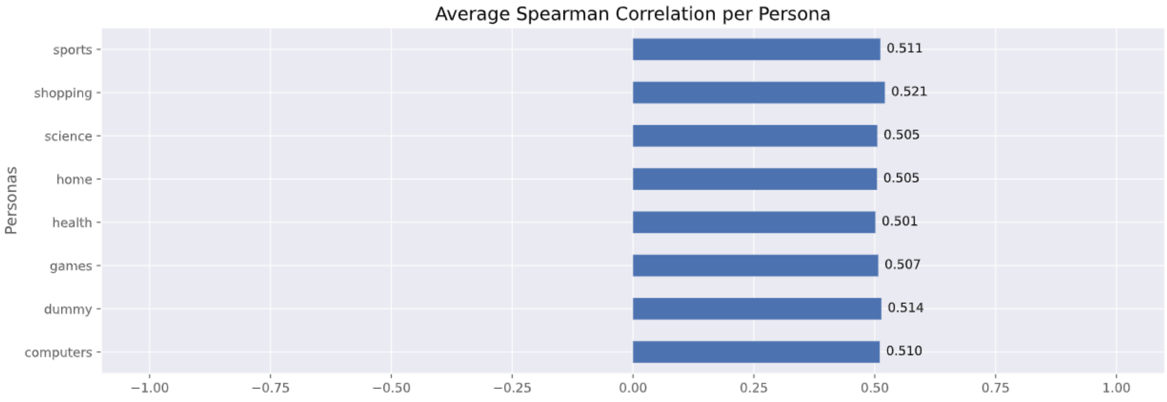


Figure 14 Average Spearman's Rank Correlation Coefficient per Persona

From the plot above, we notice that the obtained coefficient is very similar to all the personas. We can also interpret that, on average, the price influences the choice of the recommended seller, but it's not always the offer with the lowest price that is going to be the one recommended.

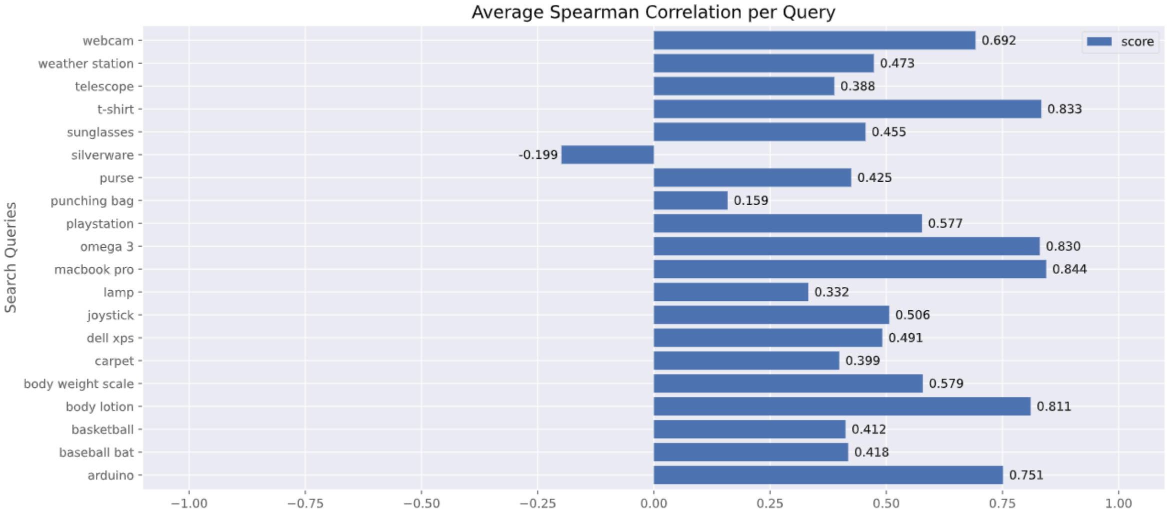


Figure 15 Average Spearman's Rank Correlation Coefficient per Query

The plot above also shows the Spearman's Rank Correlation Coefficient, but this time, per query. The first thing that stands out in this graph is that the search query "silverware" is on the negative spectrum of the coefficient, which means that Amazon tends to recommend sellers with higher prices for this search query. Other than that, we notice that for queries such as "macbook pro", "omega 3", "t-shirt", "body lotion" and "arduino", Amazon tends to order the recommended sellers by the selling price (lowest first).

6.3.2 Random Forest Model

So far we managed to prove that Amazon's buy-box order doesn't depend that much on the persona - given the results of figure 14 - and that it depends on features other than only the selling price.

The goal here was to try to create a supervised learning model that would be able to guess the product offer that amazon would recommend. To do so, we decided to build a Random Forest Classifier that took into account the the following features - which we believed were the most influential ones - of each offer:

Features
prod_condition
seller_rating
seller_n_reviews
prod_price

Table 8 Features of the Random Forest Classifier

Once we ran our model, we were able to match the recommended seller 80% of the time. The model, Gave us the following importance for each feature:

Features	Importance
prod_condition	0.007
seller_rating	0.150
seller_n_reviews	0.369
prod_price	0.474
total	1

Table 9 Importance of the Features of the Random Forest Classifier

7. Conclusion

Recent research on the impacts of user information for online shopping personalisation has provided a more complete understanding of how the modern era of almost free flow of data affects daily habits. Our findings suggest that pricing strategies relying on user profile are existent, based mainly on price steering techniques (although a certain degree of price descrimination can be observed). The obtained data reveals clear differentiation in products displayed and product display order for a given search query, based on the user's simulated profile. Despite the significant result, strategies implemented by platforms are complex, dynamic and not easy to detect, demanding a continuous effort to keep track of the ever growing number of personalisation methods.

Further analysis on the construction of the buy-box seems promising as a way of identifying new trends on the use of personal information to direct user actions. This domain reveals itself very relevant given the amount of data produced even by an average internet user, and potential traces that can be used to build a digital profile accountable for possible differentiation on shopping processes.

8. Bibliography

1. Hannak, A., Soeller, G., Lazer, D., Mislove, A. and Wilson, C., 2021. *Measuring Price Discrimination and Steering on E-commerce Web Sites | Proceedings of the 2014 Conference on Internet Measurement Conference*. [online] Dl.acm.org. Available at: <https://dl.acm.org/doi/10.1145/2663716.2663744> [Accessed 17 June 2021].

2. Chen, L., Mislove, A. and Wilson, C., 2021. An Empirical Analysis of Algorithmic Pricing on Amazon Marketplace. [online] Personalization.ccs.neu.edu. Available at: <https://personalization.ccs.neu.edu/static/pdf/amazon-www16.pdf> [Accessed 19 June 2021].