Allegro Summer e-Xperience: Intern - Data Sciencist task

Jakub Kosterna

April 29, 2021

1 First reflections

- *olist geolocation* dataset contains information about cities' locations in latitude & longitude. I guess I'll definetely use them in my algorithm - probably residents of towns next to each other (and states) have closer preferences and needs. A smart solution would be to prepare a ranking of the nearest towns for selected cities, by geographic distance.

- We must remember that Brazil is a very large and culturally diverse country. The 26 states clearly differ in many respects, as does the same for cities, of course.

- Given the lack of input in terms of time, I will not take into account time data. Of course, these are of great importance - for example, I'm quite sure that many orders are placed in December and they belong to a specific product class (see: Christianity in Brazil). However, our output function takes no time as an argument in any way.

- *freight value* seems to be an insignificant variable and too dependent on basic price, which I will definitely not take into account.

- The same applies to *payment type* and the related *payment value*.

- I think *review score* is worth including. By default, I probably will not recommend products below 4 - the customer is definitely less willing to buy them, and even when he buys them... they can potentially discourage people from visiting the website due to their quality.

For each zip code, the average distance and latitude were calculated - I decided to do this because the input of our main function does not contain detailed coordinates, but it is still valuable information, which includes regions similar to each other culturally and in terms of needs. However, the zip code already covers a certain group and is naturally closely related to the city and state.

Regardless of the input data, 10 items will be returned, all with an average grade of at least 4.5.

2 Solution

I chose not to generate one complex solution with a lot of data, e.g. in the form of a black box model. Instead, I will try to look at the problem from different angles and complete 10 products in different ways. I believe that such diversity may result in a greater probability that the customer will see the product they decide to buy.

2.1 Algorithm for the given arguments

After analyzing the available data, I decided that my function will return 10 products according to the scheme:

A: 1 product: bestseller in the city, but not belonging to the three user's favorite categories

B: 3 products: bestseller in the state, but not in the city and 5 nearest cities with at least one sold item of the category

C: 3 products: for every category, bestseller in the 5 nearest cities with at least one sold item of the category

D: 3 products: for every category, bestseller in the city (if not having any of the category in the city - take extra from C)

2.2 Algorithm without any given arguments

I step: For a user who is not logged in, we can prepare a different recommendation interface - to encourage him more, all 10 recommended products will have a verbal (not very long) user review. We will also limit ourselves to only the best products - with an average of only 5.0 (i.e. the highest possible).

II step: In order to display it nicely and at the same time to subconsciously encourage to continue browsing - we will display products with reviews between 20 and 50 characters long.

III step: The same goes for the length of the name - let's only display those with the name between 30 and 60 characters.

IV step: Let the user be able to comfortably scroll through at least two photos!

V step: Naturally, we will recommend products that have enjoyed some popularity. We only filter those that have sold at least 5 times. It will also prove that the seller is credible - it says about his systematic nature and effects on the website.

VI step: Our client can stay all over Brazil. To increase the chance that he will see a product that was just bought in his city (which may make him more likely to look at it), we can remove a bit less rows for cities where the remaining products are the most.

After all these filters, we end up with just 75 very attractive products that meet all of the above conditions. Our function will simply return a random 10 of them - by default triggering a new draw each time the user page changes. Thanks to this, the longer the customer browses the website, the more attractive offers will be found (up to 75). It will be diverse, but at the same time the subconscious effect of liking the familiar should also be triggered - it is very possible that after seeing the product for the second or third time by the user, not necessarily consciously, the user will be more likely to buy it. See: Mere-exposure effect

3 Product lists

As generated in *.ipynb*:

Case 1 - (cama_mesa_banho, papelaria, fashion_calcados), (sao paulo, SP):
- f1c7f353075ce59d8a6f3cf58f419c9c
- 99a4788cb24856965c36a24e339b6058
- 5411e9269501a870cabf632f05655131
- a2c75a23c2f838881dd4275c0cec519f
- f1c7f353075ce59d8a6f3cf58f419c9c
- e03102efbc2229024c89be731f0aedcb
- ac5e164e2eda939ffa46593f90077f9a
- 99a4788cb24856965c36a24e339b6058
- 5411e9269501a870cabf632f05655131
- ac5e164e2eda939ffa46593f90077f9a

Case 2 - (esporte_lazer, moveis_decoracao, telefonia), (rio de janeiro, RJ):
- 71a5f1c2a5fd9889ef26b5ac22aec9c6
- c6336fa91fbd87c359e44f5dca5a90ed
- aca2eb7d00ea1a7b8ebd4e68314663af
- 97017430754804328eb9597b7f85da03
- 054515fd15bc1a2029f10de97ffa9120
- 569ffd16f8032478cbeb9800f2e94ba0
- 8c5876b1c7768217964f353bc7e64393
- c6336fa91fbd87c359e44f5dca5a90ed
- aca2eb7d00ea1a7b8ebd4e68314663af
- 97017430754804328eb9597b7f85da03

Case 3 - no input data:
- bee2e070c39f3dd2f6883a17a5f0da45
- 0aabfb375647d9738ad0f7b4ea3653b1
- f1c7f353075ce59d8a6f3cf58f419c9c
- c1488892604e4ba5cff5b4eb4d595400
- cfd414b4463647f58c7775eaae06893d
- d017a2151d543a9885604dc62a3d9dcc
- 629e019a6f298a83aeecc7877964f935
- aca2eb7d00ea1a7b8ebd4e68314663af
- 5411e9269501a870cabf632f05655131
- bb50f2e236e5eea0100680137654686c