Big Data Systems and Architectures

Redis & MongoDB Assignment

Team Members

Konstantina Georgiopoulou (p2822004)

Anastasios Theodorou (p2822007)

Due Date:14.03.2021

TASKS

Task 1-Redis

1.1 How many users modified their listing in January?

We created the dataframe “data ”, which contained the users with their modifications or not in January (binary data). After that, we created the Bitmap "ModificationsJanuary" with those who finally did modifications and we found that **9969** users modified their listing on this specific month.

1.2 How many users did NOT modify their listing in January?

We performed an inversion on the “ModificationsJanuary” of the previous question and after the calculations, we found that **10031** users did not modify their listing on this month.

Combine the results with the answer of 1.1. Do these numbers match the total of your users?

We can observe with the sum of the above two answers is differ from the original by 1, so they are not much with the total sum. This happens because the BITOP operations occur at byte-level increments. Redis stores at byte-level and has no way to store the length of the exact byte. Each byte has 8 bits and the result of these operations should be an integer multiple of 8 (9969+10031)

1.3 How many users received at least one email per month (at least one email in January and at least one email in February and at least one email in March)?

We created a data-frame that counted for each user and each month the times, each one of them had received an email. Then, we made three Bitmaps for each separate month, “EmailsJanuary”, “EmailsFebruary” and “EmailsMarch”, to find those emails that each user received. Finally, we joined those Bitmaps and found that **5085** users received at least one email per month.

1.4 How many users received an email in January and March but NOT in February?

We created a bitmap containing a combination of emails received in January (“EmailsJanuary”) and in March (“EmailsMarch”) and we took the inversion of emails that have been sent in February from bitmap “EmailsFebruary”. Then we combined the above and we counted the records and found that **no** users received an email in January and March but not in February.

1.5 How many users received an email in January that they did not open but they updated their listing anyway?

From the given data we selected those that have been received in January, but not opened ("EmailsNotOpenedJanuary"), we counted those records and joined them with those from the first question

(users that have modified their listing on this specific month). So, we found that **2807** users received an email in January that they did not open but they updated their listing anyway.

1.6 How many users received an email on January that they did not open but they updated their listing anyway on January OR they received an email on February that they did not open but they updated their listing anyway on February OR they received an email on March that they did not open but they updated their listing anyway on March?

We followed the same steps as we did in the above 1.5 task. That is why, we created two more bitmaps ("EmailsOpenedFebruary", "EmailsOpenedMarch"). Next, we found that **7221** users received an email either in January, in February or in March that they did not open, but they updated their listing anyway(assigning the "or" logical operator in those three bitmaps).

1.7 Does it make any sense to keep sending emails with recommendations to sellers? Does this strategy really work? How would you describe this in terms a business person would understand?

17.99%: the percentage of those who received an email, did not open it and despite that they updated their listing (in January or in February or in March) - from the 1.6 task

36.63%: those who received an email, opened it and did not updated their listing (in January or in February or in March)

36.58%: those who received an email, opened it and updated their listing (in January or in February or in March)

From the above, we can infer that with small differences the percentage of those who received an email, opened it and not updated their listing is higher than all the other occasions. So, we would suggest that keep sending emails may be not a good idea because the updating of the listing for each user probably is not exactly depending on the receiving or not relevant emails. One business person should examine other factors as well, that affect those updatings. That is why because we can observe from our results that if a user wants to update his listing maybe will not take into consideration the email that has been sent to him, and probably he will do the updating anyway.

TASK 2-MongoDB

2.1 Add your data to MongoDB.

First of all, we created a text file with all the paths of the interested json files with the help of Windows PowerShell. Next, we read the above file in R and we made a list containing the contents of every json file, which in our case was an ad for selling a motorcycle. After that, we did some cleaning of the given data as it is described inside the R code file. In general, we dropped some fields that either added no further information to the user or contained duplicated data. Also, we converted some fields so as to be numeric or even boolean, in order to do calculations more efficiently. Finally, we added a boolean field which is true when an ad does not contain any price, or it contained an unexpectedly small price (e.g smaller than 40€) and false in all the other cases. Then, we manipulated the data in a way that can be inserted in the mongo database (through a vector).

2.2 How many bikes are there for sale?

By applying the count() function we can conclude that there are **29701** biles for sale.

2.3 What is the average price of a motorcycle (give a number)? What is the number of listings that were used in order to calculate this average (give a number as well)? Is the number of listings used the same as the answer in 2.2? Why?

By doing the aggregations, we can understand that the average price is **2962.701€**. The number of listings that are used for this calculation is **29701**.

2.4 What is the maximum and minimum price of a motorcycle currently available in the market?

We selected those fields that bikes had a price and those with a meaningful price (e.g above 40 €). So, from them, we selected the maximum and minimum price (**89000€ and 50€ accordingly**).

2.5 How many listings have a price that is identified as negotiable?

We selected those fields in the section "metadata.model" which contained the phrase "Negotiable" and after that, we counted the records to find that **1348** models had a negotiable price.

2.6 (Optional) For each Brand, what percentage of its listings is listed as negotiable?

For the purpose of this task, a new collection had been created, which contained all the brands grouped by each other and next to each one of them the number that appears in the data. Next, as we did in the previous task and here we found the negotiable models and then we grouped them by the brand name counting them as well. After that, we joined this result with the above newly created collection, in order to have the total times each brand

appears and those that are referred as negotiable. So, to sum up, we calculated the percentage of each brand's listing which is listed as negotiable and we rounded the result in two decimals.

2.7 (Optional)What is the motorcycle brand with the highest average price?

The **Semog brand** is the one with the highest average price (15600€). This result was found after calculating the average price of each brand and sorting this output in descending order. So, we selected only the first row which was the brand we were looking for.

2.8 (Optional) What are the TOP 10 models with the highest average age? (Round age by one decimal number)

For this task, we had to split the "registration date" into two parts, because it was in the form: "month / year". After that, an array was created and that is why we had to select only the last part of this array (the year). We trimmed it in order to remove the whitespaces and we converted it to integer. Later, we subtract this result with the current year (2021) and for every model we found each average age. Finally, we rounded it by one decimal and then we sorted these records in descending order to take the 10 of them.

2.9 (Optional) How many bikes have "ABS" as an extra?

From the field "extras" we counted the bikes that contained the word "ABS" and found that **4025** out of 29700 contain this feature.