Lab 2

-

Association rules

Jönköping University

Adrien Guezennec

Introduction

You can clone my repo Github to get the code directly.
https://github.com/adrienGzc/data-science-lab2

Or in a terminal: *git clone git@github.com:adrienGzc/data-science-lab2.git*

Implementation

Question 2:

There were **29363 products** sold during the week the data was recorded and **120 unique products** which make **17970102999144312104105213735180352079856680134172904172 50 rules** if we proceed with the brute force methodology. It would be suicide...

Application

Question 4, 5, 6:

To define the parameters for the apriori function I lead my choice with the amount of purchase for a product. So, for the support, I choose at least 0.006 since I thought of a minimum purchase of 50 per week (the data is based on a week so 7 times 7 make 49 which I round to 50 and divided it by the amount of purchase: 7500 -> 0.006666666, etc...). For the confidence, I decided to go with 0.3 to eliminate rules that are not relevant. The minimum length is 2 because we want to look at rules between 2 products to give the best strategy. The last one is the lift. I chose 3 because the two products have to be related in a minimum way. I do some extra filtering to delete duplicate rules but reversed and also the rules with no value.

```
Question 4, 5, 6:
Number of rules:  5

[['ground beef -> herb & pepper',
  '0.015997866951073192',
  '0.3234501347708895',
  '3.2919938411349285'],
 ['ground beef -> spaghetti',
  '0.008665511265164644',
  '0.31100478468899523',
  '3.165328208890303'],
 ['shrimp -> mineral water',
  '0.007199040127982935',
  '0.30508474576271183',
  '3.200616332819722'],
 ['tomatoes -> spaghetti',
  '0.006665777896280496',
  '0.3184713375796179',
  '3.241053850607991'],
 ['ground beef -> mineral water',
  '0.006665777896280496',
  '0.39062500000000006',
  '3.975682666214383']]
```

These settings got me 5 main rules. They basically represent the probability of A in B, so if we took the first one, people we bought ground beef are also attempting to buy herb & pepper. In this particular rule we got a support of 0.01599..., a confidence of 0.3234... (which I like to represent as 32.34% of the purchased containing ground beef contain herb & pepper) and a lift of 3.29....
So, if I had a grocery store, for example, I would put them close to each other.

Visualization

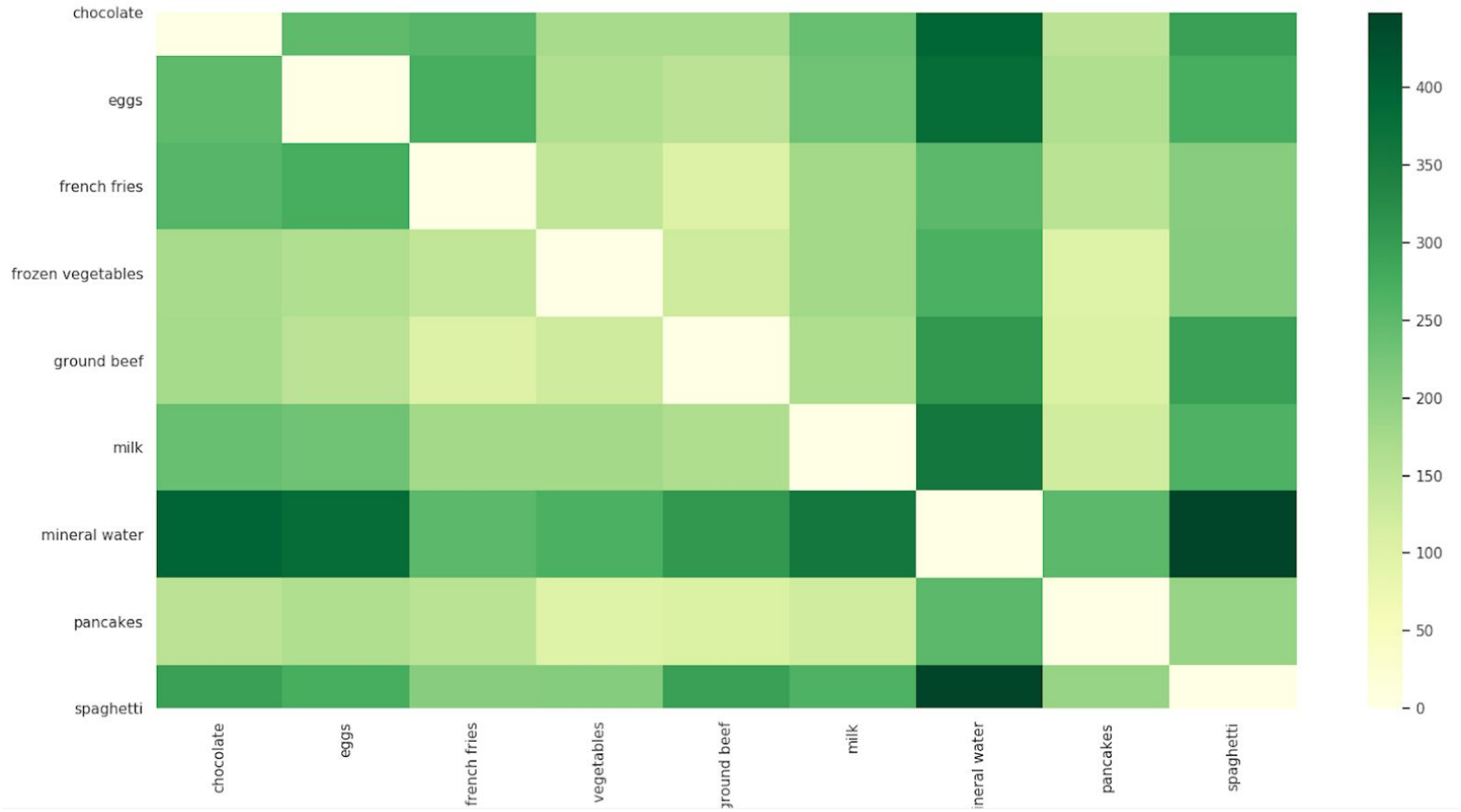
Question 7:

Question 7:

Rules	Support	Confidence	Lift
ground beef -> herb & pepper	0.0159979	0.32345	3.29199
ground beef -> spaghetti	0.00866551	0.311005	3.16533
shrimp -> mineral water	0.00719904	0.305085	3.20062
tomatoes -> spaghetti	0.00666578	0.318471	3.34105
ground beef -> mineral water	0.00666578	0.390625	3.97568

Even though there are not that many rules to analyze, I put them on a table to read it faster and quicker. So here I'm gonna take the first rules and describe what the value is and what they mean.
Support: correspond to the ratio of the transaction containing this article. So the number of transactions containing this article divided by the total of transactions.
Confidence: this indicates the ratio of element 2 in element 1 in the rules. So, for the first rule, this indicates the percentage of herb & pepper in a transaction with ground beef.
Lift: indicate that herb & pepper is 3.29199 times more likely to be bought when ground beef is bought.

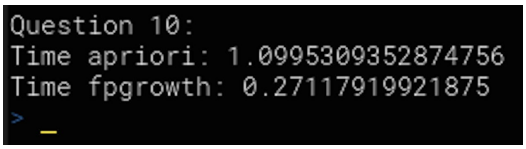
Question 9:



For the heatmap, I actually got some trouble. I had to change my 2D array for the heatmap to use the NumPy array. I had a list of list but the powerful method in NumPy help me a lot to remove the co-occurrence unneeded.
After the changes, I was able to remove those occurrences and I put my threshold to 250.
Has we can see, this confirms the rules that we got and also reveal an important point with the mineral water. Mineral water is very often purchased but I don't think this is relevant because it's a natural need that everybody purchased, so not useful to analyze.

Question 10:

After some research for fp-growth implementation, I finally reach to run it against apriori algorithm. First of all, I had to reshape a little bit the data because the algorithm only accepts a certain type of input on the list. Reshaping done, I find more accurate to run the apriori algorithm against fp-growth from the same library and not from another library.
For that, I used the same parameters used before and the time difference is quite big.



So, we got almost 0.8 milliseconds difference for the computation time. My laptop, not the best, might have an impact on the difference but it also means that both of the algorithms had been impacted. I read some documentation about the two algorithms and I found out that fp-growth is the drop-in replacement for apriori. Why? Because he is definitely less greedy in memory and he got only two loop laps in the dataset: the first lap, it builds a compact data structure called the FP-Tree and in the second lap it directly extracts the frequent itemsets from the FP-Tree.
Apriori is different, he makes multiple scans in the dataset for each candidate. Which makes him greedy in memory and slow for the computation.

Analytical and innovation part

Question 1:

For the algorithm recommendation, I would personally go with the nearest-neighbor-based methods like KNN. This algorithm is based on the similarity and distance between two instances. It's a very simple algorithm to implement or use and it is important to keep it simple in business. A neural network in deep learning would be awesome and interesting but we also have to take into consideration the context of data science and that's why it is useful to make the simplest when you can. Besides that, the limit would be with a huge dataset as the knn can be slow for the computation with a lot of instances in the dataset.

Question 2:

As I already explained above in another question, what could be improved in the dataset is to add the price of each product. Because we only based our choice on a number and the ratio of this one.

Question 4:

In a question above I suggest the use of the association rules for a grocery store. This can be pushed a little bit further and we can imagine that if two products highly related are close to each other this means that if one is almost gone than the other one could be as well. There is also all the marketing aspect to take into consideration, this means all the commercial campaign related to products (as we saw in the last assignment lab 2 - clustering).