Market Basket Analysis Project Report

July 10, 2024
Arina Lopukhina (17169A)

"I/We declare that this material, which I/We now submit for assessment, is entirely my/our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my/our work, and including any code produced using generative AI systems. I/We understand that plagiarism, collusion, and copying are grave and serious offenses in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me/us or any other person for assessment on this or any other course of study."

1. Introduction

Market Basket Analysis is a technique that allows to explore relationships among various entities and determine whether any co-occurrence is present between them, thus helping to uncover trends and patterns in a given dataset. More specifically, it allows defining frequent item sets of various sizes, which are formed from separate items stored across baskets and are often associated with each other.

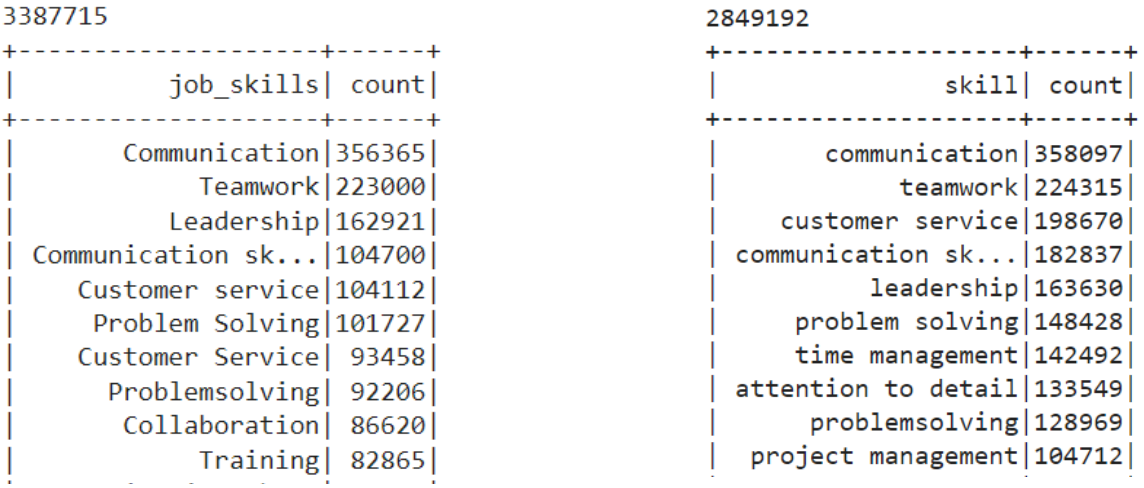
The project attempts to recreate the Apriori algorithm from scratch with the help of PySpark Python library and then use it for analyzing a LinkedIn dataset consisting of job listings. The Apriori algorithm has been chosen thanks to its simplicity yet high effectiveness when it comes to frequent itemsets search. The goal of this exercise is to locate the most important skills required by employers, since if a skill is frequently mentioned, it likely indicates its high significance for a given job type. Additionally, the algorithm results help to understand if there are any skills that are interdependent and appear in the job listings together.

This report provides an overview of the successful algorithm implementation, describing the whole process from EDA and data preprocessing to the algorithm results discussion and reflection.

2. Data exploration

The initial dataset includes two columns - one containing the job listing LinkedIn link, and the other column is represented as lists enumerating the skills requested for a given job. The whole dataset contains 1,294,374 rows. To better understand the included skills, I have broken down each list into separate words and then counted instances of those words and grouped the counts by unique skill. Without any additional text manipulations, I have received a table containing 3,387,715 unique skills, with Top-3 being communication, teamwork, and leadership.

However, there were instances of repeating words due to spelling errors or differences in cases. Unfortunately, my attempts to apply fuzzy string matching was hard to implement due to the computational limitations. So, I have decided to simply convert all the strings to lowercase as an attempt to generalize some of the writing differences. This step significantly reduced the number of unique skills - it went down to 2,849,192, thus removing over 500 thousand 'duplicates'. Nonetheless, the top-mentioned skills remained roughly the same - the only significant change was that 'leadership' was no longer in the Top-3, and 'customer service' became more frequent instead. Afterwards, I applied lowercasing to the entire dataset to use it for future analysis.



```
3387715
+--------------------+------+
|          job_skills| count|
+--------------------+------+
|       Communication|356365|
|            Teamwork|223000|
|          Leadership|162921|
| Communication sk...|104700|
|    Customer service|104112|
|     Problem Solving|101727|
|    Customer Service| 93458|
|      Problemsolving| 92206|
|       Collaboration| 86620|
|            Training| 82865|
```

```
2849192
+--------------------+------+
|               skill| count|
+--------------------+------+
|       communication|358097|
|            teamwork|224315|
|    customer service|198670|
| communication sk...|182837|
|          leadership|163630|
|     problem solving|148428|
|     time management|142492|
| attention to detail|133549|
|      problemsolving|128969|
|  project management|104712|
```

Figure 1: Initial Top-10 Items (left) versus Lowercase (right).

Next, due to the computational constraints, it is recommended to take a sample from the dataset to ensure the convergence of the algorithm. As per the student's book, *Mining for Massive Datasets*, it is common to choose a sample size of 1% from the total dataset, which is what I did for my project (Leskovec J, Rajaraman A, Ullman JD). The further data processing and algorithm application is performed on the 1%, namely, on 13,057 rows randomly extracted from the original dataframe with lowcasing performed.

3. Preparing the baskets

To create proper ‘baskets’, I converted the skills column into an RDD, resilient distributed dataset, which is a Spark data structure for addressing big data by partitioning it and splitting across different clusters. Then, I received roughly 13,000 sets of skills in an RDD form, which will help me define the support level. Among the baskets, the maximum number of skills per basket is 171, with an average of 21 skills per basket.

```
[['employee training',
  'food safety',
  'inventory management',
  'team management',
  'kitchen supervision',
  'food quality control',
  'food safety',
  'cost control',
  'menu management',
  'hiring',
  'staff scheduling',
  'sales management',
  'profit optimization',
  'customer service',
  'guest experience',
  'service area organization',
  'teamwork',
  'hsd or ged',
  'passion for helping and serving others',
  'leadership experience',
  'motivation',
  'development of others',
  'commitment to excellence',
```

Figure 3: RDD view.

Finally, I set the support level of the sample - hence, to be considered frequent, an item needs to exceed the support threshold of a given value (appear in a number of baskets which is higher than the support level). I have experimented with setting the support to 1% and 2% of the sample (131 and 262 occurrences respectively).

As the next step, I needed to replace strings with corresponding hash values which helps reduce storage needed, as the string values get mapped and replaced by numeric values which require less space. As a result, my baskets began to look like the following set of numbers:

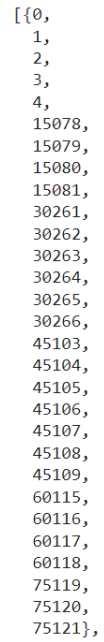


Figure 4: Hashed values view.

Finally, the preprocessing part is complete and the data is ready to be fed into the algorithm.

4. Algorithm execution

The first step is creating a Map which takes the hash value and, for every basket item, it creates a key-value pair by using the hash index as a key and assigns a value of 1. Hence, we receive pairs looking like this - (hash key, 1). Then, the Reduce part of the function sums up all the 1's for a given hash index value, which allows us to have a number of occurrences per given item. Lastly, I filter only for values that exceed the support threshold, and hence it allows me to identify the genuinely frequent skills, count them, and store them together as well. In case no singletons exceed the threshold, the algorithm stops and signals that the support level is too high. After the first step, we can see that 'communication' is the most frequent string, which we have already noted during the preliminary data analysis. With the support equal to 1%, I received a total of 191 frequent items, while with the higher threshold value (of 2%), there were only 81.

In the second step, I proceed to identify items which frequently appear together. From the monotonicity assumption, 'if a set I of items is frequent, then so is every subset of I', which helps reduce the amount of processing required in the future iterations as I would only consider those items which made it over the threshold in the first pass (Leskovec J, Rajaraman A, Ullman JD). To detect candidate pairs, the algorithm takes each basket and looks at all the possible combinations of size k (in case of pairs, k=2), then creates a key-value pair, assigning each

combination a value of 1. Finally, the Reduce step calculates the value sum and allows us to finally find out which two frequent items have appeared together the most. In both cases, it is a combination of ‘communication’ and customer service’.

Similarly, the algorithm proceeds to examine triplets, quadruples, and so on until all the options are exhausted. In the case of higher support value, there were no combinations beyond triplets, while lowering support value allowed to identify also a few quadruplets, as can be seen below:

```
Frequent singletons
Number of frequent singletons: 191
Most frequent one: communication

Sets of: 2
Frequent sets of 2 : 252
Most frequent set of 2 : ['customer service', 'communication']

Sets of: 3
Frequent sets of 3 : 100
Most frequent set of 3 : ['teamwork', 'customer service', 'communication']

Sets of: 4
Frequent sets of 4 : 26
Most frequent set of 4 : ['teamwork', 'customer service', 'problemsolving', 'communication']
```

Figure 5: Algorithm Results with 1% Support Threshold.

```
Frequent singletons
Number of frequent singletons: 81
Most frequent one: communication

Sets of: 2
Frequent sets of 2 : 69
Most frequent set of 2 : ['customer service', 'communication']

Sets of: 3
Frequent sets of 3 : 22
Most frequent set of 3 : ['teamwork', 'customer service', 'communication']

Sets of: 4
No frequent sets of k 4
Stop here.
```

Figure 6: Algorithm Results with 2% Support Threshold. .

5. Results discussion

The algorithm has performed successfully and allowed me to derive meaningful insights about the dataset. It appeared that the most valuable skill is communication, and the most frequently combined items include communication, teamwork, customer service, and problem solving. These skills are core to most types of jobs across different fields that presume working and collaborating with other people, so the results are not totally surprising, as we live in a society after all.

Among the parameters that affected the algorithm performance, setting higher support has significantly speeded up the calculations, as the processing time went from over half-an-hour to just 9 minutes. However, this also resulted in a partial loss of results as we saw that with the 2% support level the algorithm did not manage to find any combinations past triplets. It is rather a matter of prioritization - perhaps, in certain cases it is crucial to identify all the possible combinations, but with my scope, adding the information about quadruples was nice-to-have, but having an algorithm converge in a reasonable timeframe was more important. Given the time constraint, as a continuation of this project, it would be interesting to compare how much time PCY or other alternative algorithms would take to process this data as Apriori is known to be less efficient.

Another factor that played a meaningful role in performance optimization was the data preprocessing step. Modifying the skills to lowercase format has reduced the number of unique items by nearly 20%, thus saving us time and effort by eliminating items that contain practically the same information, which allowed the algorithm to converge several minutes faster. I suppose that with more text preprocessing, the performance could have been improved further as there was a fair share of items containing several words or even full sentences, which could be synthesized to a synonymous word or two, hence potentially altering the frequency counts. As, unfortunately, I still haven't taken professor Ferrara's Text mining course, I did not dive deeper into this matter, but it would be very interesting to consider it as an additional action item.

6. Conclusion

Overall, the dataset analysis and the implementation of Apriori algorithm was a success, despite the limitations described above. It has definitely helped me visualize the processes behind Market Basket Analysis on real-world data, and I would be happy to experiment further in order to improve the performance as well as make the solution more scalable.

References:

1. 1.3M LinkedIn jobs & skills (2024). Kaggle: Your Machine Learning and Data Science Community.https://www.kaggle.com/datasets/asaniczka/1-3m-linkedin-jobs-and-skills-2024
2. Bertolotti F. MarketBasket Notes. GitHub. https://github.com/f14-bertolotti/labs/tree/main/DSE/MarketBasket
3. Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). Frequent itemsets. In Mining of massive datasets (pp. 191-227). Cambridge University Press.