Market-Basket Analysis

Algorithms for Massive Data Project

Tatiana Getling
tatiana.getling@studenti.unimi.it
Matricola 963368

I declare that this material, which I now submit for assessment, is entirely my 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 work, and including any code produced using generative AI systems. I understand that plagiarism, collusion, and copying are grave and serious offences 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 or any other person for assessment on this or any other course of study.

Introduction

This project aims to implement the Apriori algorithm to identify frequent itemsets for market-basket analysis of the LinkedIn Jobs&Skills dataset. The objective is to explore how often certain skills appear together and what skillsets are in-demand in the job market.

The report is structured into several chapters. The first chapter explores the dataset and details the preprocessing steps that were taken, including eliminating irrelevant data, handling missing values, and standardizing the data format to ensure consistency.

The second chapter introduces the algorithm selection, provides theoretical background relevant for the implemented algorithm, and explains the reasoning behind choosing Apriori as the main algorithm for this task.

In the third chapter we discuss the implementation of the algorithm and the difficulties encountered during the experiments. We describe several aspects that were taken into consideration to ensure the optimal performance and scalability of our implementation. We further analyze the results obtained from the experiments and assess the final performance of our algorithm.

In the final chapter we draw conclusions based on the results of our experiments, discuss the limitations and possible improvements for our approach.

Data Preparation

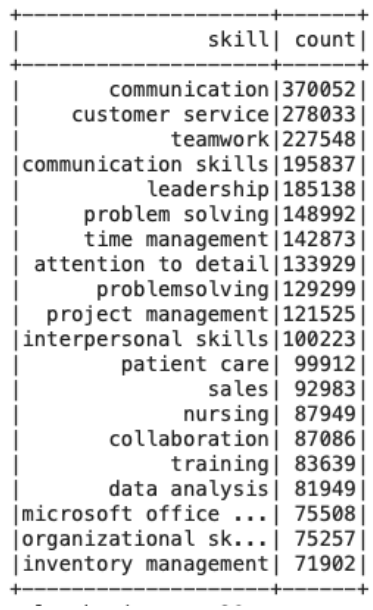
The analysis was performed on the dataset sourced from Kaggle (Asaniczka, 2024). The dataset, «LinkedIn Jobs & Skills», consists of three csv files containing the data for the job skills, job summary and job postings. Given the aim of the project we have removed the irrelevant csv files from the project environment to focus exclusively on the job skills table. The job skills dataset consists of two columns: «job_link» (containing links to specific job posting) and «job_skills» (containing the list of the skills required for the position).

The first step in the preprocessing of the data was to read the relevant CSV file into a Spark DataFrame. This allowed us to use Spark's distributed computing capabilities to handle the dataset more efficiently. We have also eliminated the null values from the dataset, as they are irrelevant for our project's objective. The initial exploration of the data included displaying a summary of the most frequent skills and counting the total number of distinct skills. This step provided an overview of the dataset's content and helped identify any immediate issues with the data. The first overview allowed us to notice the inconsistent spelling of the job skills: the skills with uppercased and lowercased spelling are considered separate skills within the dataset, hence the further count of the data could be inaccurate.

skill	count
Communication	368202
Teamwork	226205
Leadership	184292
Customer service	166158
Communication skills	116169
Customer Service	110400
Problem Solving	102020
Sales	92718
Problemsolving	92489
Nursing	87419
Collaboration	86774
Training	83178
Project Management	81080
Communication Skills	78700
Attention to detail	75448
Microsoft Office ...	73351
Time management	72460
Time Management	69752
Scheduling	64081
Microsoft Office	60260

Fig.1 - The skills count before processing

In order to avoid inconsistent data, we have proceeded to lowercase all the values and printed the top rows of the resulting table to compare the results.



skill	count
communication	370052
customer service	278033
teamwork	227548
communication skills	195837
leadership	185138
problem solving	148992
time management	142873
attention to detail	133929
problemsolving	129299
project management	121525
interpersonal skills	100223
patient care	99912
sales	92983
nursing	87949
collaboration	87086
training	83639
data analysis	81949
microsoft office ...	75508
organizational sk...	75257
inventory management	71902

Fig.2 - The skills count after lowercasing the values

This step has shown that the count for the skills has changed drastically compared to the unprocessed data (the overall skill count went from 3298454 to 2770596).

Further examination revealed that some skills were duplicated due to inconsistent use of spaces (e.g., "problem solving" vs. "problemsolving"). To eliminate these inconsistencies, several techniques were considered, but taking into account the size of the dataset and the final objective of the project (i.e. revealing the frequent itemsets) it was decided to handle these values in a simpler way that wouldn't require too much computational power, specifically eliminating spaces in all of the values for the job skills altogether, thus removing the duplicate values.

By comparing the top rows of the resulting table, we can once again notice how the frequency of certain skills has changed compared to the previous tests.

skill	count
communication	370052
problemsolving	278295
customerservice	278069
teamwork	245192
communicationskills	195844
leadership	185139
timemanagement	143255
attentiontodetail	133971
projectmanagement	121540
interpersonalskills	100229
patientcare	99926
sales	92983
nursing	87949
collaboration	87086
training	83639
dataanalysis	81958
microsoftofficesuite	75544
organizationalskills	75261
inventorymanagement	71904
highschooldiploma	67357

Fig.3 - The skill count for the processed dataset

Once the dataframe values are consistent and standardized, we can proceed to the next step. For further implementation of the Apriori algorithm, we transformed the dataframe into a resilient distributed dataset (RDD) using Spark to facilitate distributed data processing and organized it into baskets from rows. To optimize data processing, we created a hash table of the job skill values by assigning integer values (indices) to each unique job skill. Given the limitations of the available computational resources, we also sampled a portion of the dataframe to test the algorithm on a smaller dataset, before running the final experiments on the full data.

Algorithm Selection and Implementation

The problem of frequent itemset mining can be handled using different algorithms, such as Apriori algorithm, ECLAT, FP-growth and others (Luna, Fournier-Viger, & Ventura, 2023). In the scope of this project we will be focusing exclusively on the Apriori algorithm, which is widely used for market basket analysis, thanks to its effectiveness, considerable simplicity of implementation and extensive research and literature available on the topic.

The algorithm is based on the anti-monotonicity property, stating that if an itemset is frequent, all of its subsets must also be frequent, and it works iteratively by performing the following steps:

1) Identification of the unique items present in the dataset and calculation of their frequency.
2) Selecting the the most frequent items (singletons) based on a specified threshold (minimum support).
3) Generating candidate itemsets of increasing length based on the set obtained from the previous step and repeating this step until no new candidates are found (IBM, 2024).

The parameters fed to the algorithm function include the RDD (obtained during the preprocessing steps described in the previous chapter), the hash table and the minimum support value. The minimum support value is counted based on the total number of baskets with the threshold of 0.5% (for the sample data we pass a fixed value to speed up the testing process).

At the first step of the algorithm we calculate the frequent singletons by flattening the basket list, counting the skills and aggregating the values to count the occurrence of each skill. We then filter the resulting set by leaving only the skills that have a value greater than the minimum support value. If the frequent singletons are successfully found, we proceed to count the larger itemsets based on the obtained results. After each round of the algorithm we print out the number of calculated frequent itemsets and the value (found from the hash

table) of the most frequent itemset. The algorithm function takes advantage of the spark functionality to optimize the calculation time.

```
Singletons:
Counted singletons:
26
Most frequent:  communication
Itemsets of size 2
Counted itemsets if size 2:
20
Most frequent:  ['customerservice', 'communication']
Continue counting
Itemsets of size 3
Counted itemsets if size 3:
8
Most frequent:  ['teamwork', 'customerservice', 'communication']
Continue counting
Itemsets of size 4
No more itemsets
```

Fig.4 - The results of the sample run of the algorithm

From the first run of the function we can see that the algorithm seems to be working correctly by identifying frequent itemsets of increasing size up to itemsets of size 3. These results indicate that we can proceed with testing our algorithm on the complete dataset.

Experimental Results and Discussion

To evaluate the effectiveness of our Apriori algorithm implementation we conducted the experiment on the whole dataset, consisting of 1294374 baskets. The minimum support threshold was set at 1%, which translated to 12943 occurrences.

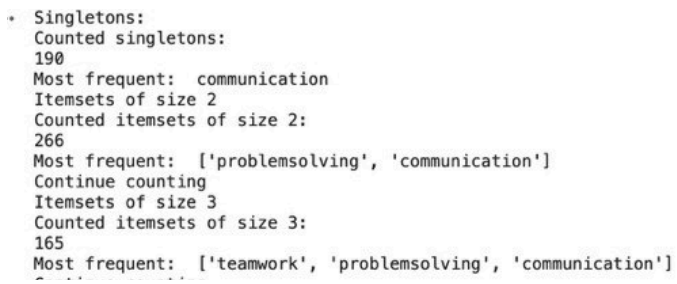


Fig.5 - The results of the complete dataset run of the algorithm

The algorithm successfully identified frequent itemsets up to size 3. The resulting most frequent values are:

- Singleton: «Communication»
- Pair: [«Problem solving», «Communication»]
- Triplets: [«Teamwork», «Problem solving», «Communication»]

As we can see, these skills frequently co-occur in the job descriptions, with the «communication» skill being the most frequent on the whole dataset as well as on the sample. It is worth noticing that the identified itemsets are consistent with our trial runs on the dataset sample, which demonstrates the robustness of our algorithm.

By analyzing these results, we can draw a conclusion, that overall soft skills seem to be prevalent in the job requirements compared to the hard skills (which are not identified as the most frequent ones). A possible continuation of this research could include analyzing the frequency of the hard skills to provide more insights into the «technical» requirements for the jobs compared to the «soft» ones.

While the experiment can be considered successful, it is important to underline some limitations of the project. Given the limited computational resources, it wasn't possible to experiment with a wider range of values (for example by setting a smaller support value) or compare the results based on the support threshold value changes.

Another potential improvement could include optimizing the algorithm performance, which, even with the spark functionality, still required significant execution time. In order to ensure the scalability of the algorithm on even larger datasets, it would be important to consider more advanced pruning strategies as well as optimizing the distributed computing processes. Furthermore, it could be beneficial to compare the results of the Apriori algorithm to other algorithms used for market basket analysis, to ensure the effectiveness of the algorithm and find more valuable insights.

Conclusion

The algorithm proved effective in identifying frequent itemsets for market basket analysis from the job skills dataset. It demonstrated consistent results for the sample data, as well as the whole dataset, thus confirming the correct implementation of such. During the research we considered future improvements and possible optimizations that could enhance algorithm's performance and scalability.

References

1. Asaniczka. (2024). 1.3M LinkedIn Jobs and Skills [Data set]. Kaggle. https://www.kaggle.com/datasets/asaniczka/1-3m-linkedin-jobs-and-skills-2024

2. Luna, J. M., Fournier-Viger, P., & Ventura, S. (2019). Frequent Itemset Mining: a 25 Years Review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9.6*

3. IBM. (2024). What is the Apriori algorithm? IBM. https://www.ibm.com/topics/apriori-algorithm