Implementation of Market Basket Analysis on LinkedIn Jobs & Skills Dataset Using Apache Spark

Niloofar Adel*

June 11, 2024

"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."

1 Introduction

This report implements and analyzes the Apriori and Multistage algorithms for market basket analysis on the LinkedIn Jobs & Skills dataset using Apache Spark, as stated in the course's project description. My objective is to find frequent itemsets, representing common co-occurrences of job skills within the job postings on LinkedIn[1]. By leveraging the parallel processing capabilities of Apache Spark, I will efficiently process and analyse the dataset to extract meaningful insights. Additionally, association rules were extracted using the Apriori algorithm to determine which skills are highly interdependent.

2 Data Structure and Preparation

2.1 The Dataset Structure

The LinkedIn Jobs & Skills dataset, which contains 1.3 million job listings scraped from LinkedIn in 2024, is a valuable resource for various research tasks including job market analysis, skills mapping, job recommendation systems, and more[1]. It comprises three CSV files: job_skills, job_summary, and linkedin_job_postings. For this project, I focused on the job_skills file, which includes two columns:

1. job_link: A URL to the job posting.
2. job_skills: A string of comma-separated skills associated with a job posting.

Table 1 provides a sample from the job_skills file.

In market basket analysis applied to LinkedIn job postings, we have three key concepts: the basket, the items, and the frequent itemsets. In the context of LinkedIn Jobs & Skills dataset, the basket represents a single job posting, containing various skills that an employer is seeking. So, each job posting is akin to a transaction in a traditional market basket analysis.

The items are individual skills listed in the job postings. And finally, Frequent itemsets are groups of skills that frequently appear together in multiple job postings. These itemsets are identified through data mining algorithms such as Apriori or Multistage, which analyze the collection of job postings to discover patterns and associations. Frequent itemsets are crucial for understanding common skill combinations that employers value. Before applying the market basket analysis algorithms, I preprocessed the dataset to prepare it for further analysis.

*University of Milan. niloofar.adel@studenti.unimi.it

Table 1: LinkedIn Jobs & Skills dataset

job_link	job_skills
https://www.linkedin.com/jobs/view/housekeeper-i-pt-at-jacksonville-state-university-3802280436	Building Custodial Services, Cleaning, Janitorial Services, Materials Handling ...
https://www.linkedin.com/jobs/view/assistant-general-manager-huntington-4131-at-ruby-tuesday-3575032747	Customer service, Restaurant management, Food safety, Training, Supervision, Scheduling, Inventory, Cost control, Sales ....
https://www.linkedin.com/jobs/view/school-based-behavior-analyst-at-ccres-educational-and-behavioral-health-services-3739544400	Applied Behavior Analysis (ABA), Data analysis, Behavioral assessment, Positive behavior support, Programming development...

2.2 Data Preparation

The preprocessing step is crucial when working with text data to ensure accuracy and effectiveness in subsequent analysis. In this pipeline, a meticulous approach was taken to handle the skills data. Firstly, skills containing the word "and" were carefully split into two separate skills, recognizing that "and" often signifies the presence of multiple skills within a single entry. This splitting was done by identifying standalone occurrences of "and" and removing it from the text, thereby preserving the integrity of each individual skill. Leveraging NLTK and STRING libraries, I then proceeded to clean the data by removing punctuations and stopwords [2][3]. Non-alphabetical characters and extra spaces were also eliminated to maintain a clean dataset. Additionally, the text was converted to lowercase to ensure uniformity and to avoid discrepancies arising from case sensitivity.

3 Methodology

The project involves implementing two key algorithms: Apriori and Multistage algorithms. Both algorithms are designed to find frequent itemsets in a dataset, but they differ in their approach and efficiency.

3.1 Apriori Algorithm

The Apriori algorithm is a classical algorithm for frequent itemsets mining and association rules learning. The algorithm is a stepwise technique that starts with the simplest rule and adds individual items to the k+1 itemset, where k sets of items are used[4].

The Apriori algorithm is based on the rule that all subsets of the frequently repeating itemset must also consist of frequently repeating sets and use an iterative approach. First, there are frequently repetitive sets with one element. This set is called L1 (frequently repeating 1-element set). L1 is used to obtain L2 (a repetitive 2-element cluster). The algorithm works repetitively to find the most repetitive sets that can be obtained. The generation of each L_k involves scanning the entire database [4].

A frequent itemset is a subset of items that occurs frequently in the transactions. By identifying these frequent itemsets, the algorithm helps uncover patterns and associations in the data, which can provide valuable insights for further analysis. There are two key concepts need to be point out:

1. **Support:** The support of an itemset measures the frequency of occurrence of that itemset in the transactions. The Apriori algorithm uses a minimum support threshold to determine which itemsets are considered frequent.

2. **Apriori Principle:** This principle states that if an itemset is frequent, then all of its subsets must also be frequent. This principle guides the algorithm's approach to efficiently generate candidate itemsets and prune infrequent ones.

Algorithm Workflow:

1. **Initialization:** The algorithm starts by identifying frequent individual items, termed L1, by scanning the dataset and counting the occurrences of each item.

2. **Generating Candidate Itemsets:** Based on the frequent itemsets of size k, the algorithm generates candidate itemsets of size k+1 by combining pairs of frequent itemsets. This step ensures that all subsets of a frequent itemset are also frequent, adhering to the Apriori principle.

3. **Support Counting:** The algorithm scans the dataset again to count the occurrences of each candidate itemset. Only candidate itemsets with a support count greater than or equal to the minimum support threshold are considered frequent.

4. **Pruning Infrequent Itemsets:** Any candidate itemset that is not frequent is pruned from further consideration, as it cannot be part of any frequent itemset.

   The algorithm will then repeat Steps 2 to 4 iteratively until no new frequent itemsets can be generated or until a predetermined maximum itemset size is reached.

3.2 Multistage Algorithm

The Multistage algorithm, also known as Multistage Apriori, is an enhancement of the traditional Apriori algorithm designed to improve efficiency by reducing the number of candidate itemsets generated during each pass. The multistage algorithm comprises three main stages:

1. **Initialization and Frequency Counting:** In the first pass, the algorithm counts the frequency of individual items and also counts the co-occurrence of item pairs using multiple hash functions. This stage identifies frequent items and buckets in hash tables.

2. **Hashing to the Second Table:** In the second pass, the algorithm hashes candidate pairs to a second hash table based on the frequent items identified in the first pass. This reduces the search space by focusing only on potential candidate pairs.

3. **Candidate Pair Counting:** In the third pass, the algorithm counts the occurrence of candidate pairs using a combination of hash functions. This step identifies frequent pairs based on the support threshold set earlier.

Overall, the multistage algorithm offers a scalable approach to identifying frequent itemsets, particularly suited for large datasets where efficiency and computational complexity are concerns.

4 Results

4.1 Frequent Itemsets Identified using Apriori Algorithm

Table 2 presents some of the frequent itemsets identified using the Apriori algorithm. Each itemset represents a combination of skills that frequently appear together in job postings. The implemented Apriori identifies all frequent singletons, doubletons and higher order itemsets if available.

Table 2: 5 most Frequent Itemsets Identified by Apriori Algorithm

Itemsets	**Support**	**Count**
{communication}	0.2862	3679
{customer service}	0.2176	2797
{teamwork}	0.1759	2261
{communication skills}	0.1574	2024
{leadership}	0.1417	1822

Support is the fraction of transactions in which the itemset appears. It is calculated as:

$$\text{support}(X) = \frac{\text{Number of transactions containing } X}{\text{Total number of transactions}}$$

For example, a support of 0.2862 for the itemset {communication} means that 28.62% of all transactions (or job descriptions) contain the skill "communication".

Figure 1 shows a word cloud of the skills identified as frequent by the Apriori algorithm. In the word cloud, the size of each skill corresponds to its frequency within the itemsets. The larger the skill, the more frequently it appears. As observed, the most frequent skills include {communication}, {customer service}, {teamwork}, etc.

Figure 1: Word could for Apriori frequent itemsets

4.2 Frequent Itemsets Identified using Multistage Algorithm

The Multistage algorithm also identified frequent itemsets with comparable support and count values. This algorithm is implemented in a way that it only identifies frequent pairs.

Table 3: Frequent Itemsets Identified by Multistage Algorithm

Itemsets	Support	Count
{'teamwork', 'communication'}	0.0876	1126
{'communication', 'customer service'}	0.0783	1007
{'teamwork', 'customer service'}	0.0602	774
{'problemsolving', 'communication'}	0.0552	710

In the context of the Multistage algorithm, support is calculated as the proportion of transactions (in this case, job listings with skills) that contain a particular itemset (a combination of skills). For example, the support value for {'teamwork', 'communication'} is 0.0876 which means that this itemset appears in approximately 8.76% of all transactions. It indicates that teamwork and communication skills frequently appear together in job listings. Figure 2 presents a word cloud of the skills identified as frequent by the Multistage algorithm.

Figure 2: Word could for Apriori frequent itemsets: The larger the skill appear in the word cloud, the more frequent it is in the itemsets.

4.3 Association Rules

The association rules generated from the frequent itemsets provide insights into the dependencies between skills. As the Multistage algorithm only outputs frequent pairs, the association rules were not available. Consequently, I only extracted association rules based on frequent itemsets identified by the Apriori algorithm. Table 4 summarizes the top 10 association rules.

The table of association rules offers valuable insights into the interrelationships among various items or concepts, derived through the application of association rule mining techniques. Each column in the table provides distinct information, aiding in the comprehensive understanding of the associations discovered.

Antecedents and Consequents: These columns delineate the items implicated in each association rule. The antecedents denote the items present prior to a specific condition or event (the consequents).

Antecedent Support and Consequent Support: These columns denote the support values corresponding to the antecedents and consequents, respectively. Support quantifies the frequency with which the antecedents and consequents co-occur in the dataset, expressed as the proportion of transactions containing both.

Support: This column reflects the overall support value for the association rule, indicating the proportion of transactions containing both the antecedents and consequents out of all transactions. For instance, The support value of 0.0119 for the association rule involving ('vision insurance', 'health insurance') and ('dental insurance') underscores the prevalence of this relationship within the dataset.

Confidence: Confidence signifies the reliability or strength of the association rule, representing the proportion of transactions containing both the antecedents and consequents out of those containing the antecedents alone. A confidence value of 0.980 for ('vision insurance', 'health insurance') and ('dental insurance') suggests a high likelihood of observing this relationship, given the presence of its antecedents.

Lift: Lift measures the strength of the association rule by comparing the observed support to the expected support under independence, with values exceeding 1 indicating a positive association. For example, with a lift value of 40.535, the association rule involving ('vision insurance', 'health insurance') and ('dental insurance') suggests a strong positive association between the antecedents and consequents.

Leverage: Leverage quantifies the deviation of observed support from the expected support under independence, offering insights into the extent of association beyond chance. Again, a leverage value of 0.011 for ('vision insurance', 'health insurance') and ('dental insurance') highlights the magnitude of association between the antecedents and consequents, surpassing what would be anticipated under independence.

Figure 3 depicts association rules explained above with a directed graph. Nodes represent items, and directed edges represent the association rules, with each edge pointing from an antecedent to a consequent.

Table 4: Assosiation Rules of Apriori Algorithm

Antecedents	**Consequents**	**Antecedent Support**	**Consequent Support**	**Support**	**Confidence**	**Lift**	**Leverage**
1. (vision insurance, health insurance)	(dental insurance)	0.012	0.024	0.011	0.980	40.535	0.011
2. (life insurance, vision insurance)	(dental insurance)	0.011	0.024	0.010	0.946	39.105	0.010
3. (word)	(excel)	0.012	0.032	0.011	0.925	28.674	0.011
4. (life insurance, dental insurance)	(vision insurance)	0.011	0.025	0.010	0.915	35.657	0.010
5. (vision benefits)	(dental benefits)	0.011	0.011	0.010	0.910	76.023	0.010
6. (cash handling, communication)	(customer service)	0.0126	0.217	0.011	0.907	4.172	0.008
7. (sales, problemsolving, communication)	(customer service)	0.012	0.218	0.011	0.904	4.156	0.008
8. (cash handling)	(customer service)	0.024	0.218	0.022	0.895	4.112	0.017
9. (dental insurance, health insurance)	(vision insurance)	0.014	0.026	0.012	0.885	34.469	0.012
10. (leadership, problemsolving, customer service)	(communication)	0.014	0.286	0.012	0.881	3.079	0.008

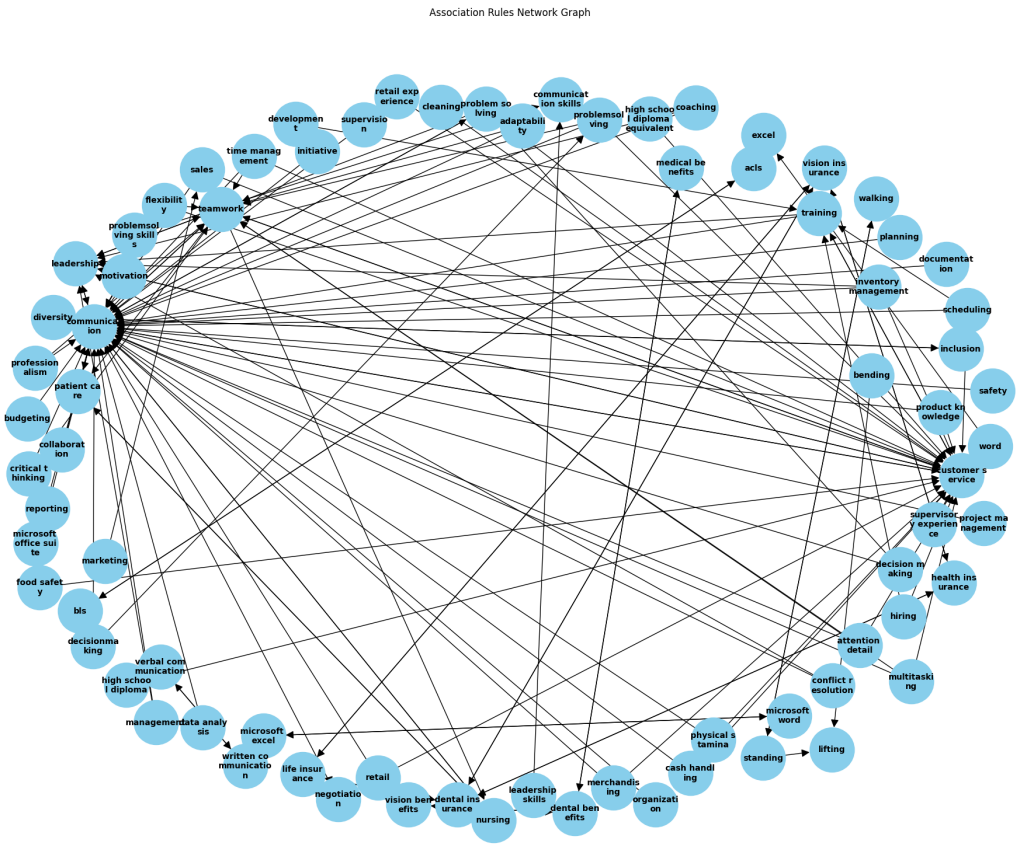

Figure 3: Association Rules Graph

5 Conclusion

In this project, I successfully implemented market basket analysis on the LinkedIn Jobs & Skills dataset using Apache Spark, leveraging both the Apriori and Multistage algorithms to identify frequent itemsets and generate association rules.

Both algorithms successfully identified frequent itemsets, but the Apriori algorithm offered a more comprehensive view by identifying frequent itemsets of varying sizes and generating association rules. The Multistage algorithm, while efficient, was limited to identifying frequent pairs in this implementation.

Overall, this project demonstrated the power of market basket analysis in extracting valuable insights from large datasets. By identifying frequent skill combinations and their associations, the analysis provides a deeper understanding of the skills demanded in the job market. This information can be beneficial for job seekers, employers, and educational institutions aiming to align curricula with market needs.

References

[1] Linkedin jobs skills dataset. Accessed: 2024-06-01.

[2] Steven Bird, Edward Loper, and Ewan Klein. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.

[3] Python Software Foundation. Python string library documentation. https://docs.python.org/3/library/string.html. Accessed: Jun. 6, 2024.

[4] Yüksel Akay Ünvan. Market basket analysis with association rules. *Communications in Statistics - Theory and Methods*, 50(7):1615–1628, 2021.